applied sciences

MDPI

Article

A Novel Object Detection Method of Pointer Meter Based on Improved YOLOv4-Tiny

Wenliang Xu [1], Wei Wang [2,*], Jianhua Ren [1,*], Chaozhi Cai [1] and Yingfang Xue [1]

[1] School of Mechanical and Equipment Engineering, Hebei University of Engineering, Handan 056038, China; caichaozhi1983@163.com (C.C.)
[2] Office of Academic Research, Hebei Finance University, Baoding 071051, China
* Correspondence: wangwei@hbfu.edu.cn (W.W.); renjianhua@hebeu.edu.cn (J.R.)

Abstract: Pointer meters have been widely used in industrial field due to their strong stability; it is an important issue to be able to accurately read the meter. At present, patrol robots with computer vision function are often used to detect and read meters in some situations that are not suitable for manual reading of the meter. However, existing object detection algorithms are often misread and miss detection due to factors such as lighting, shooting angles, and complex background environments. To address these problems, this paper designs a YOLOv4-Tiny-based pointer meter detection model named pointer meter detection-YOLO (PMD-YOLO) for the goal of practical applications. Firstly, to reduce weight of the model and ensure the accuracy of object detection, a feature extraction network named GhostNet with a channel attention mechanism is implemented in YOLOv4-Tiny. Then, to enhance feature extraction ability of small- and medium-sized targets, an improved receptive field block (RFB) module is added after the backbone network, and a convolutional block attention module (CBAM) is introduced into the feature pyramid network (FPN). Finally, the FPN is optimized to improve the feature utilization, which further improves the detection accuracy. In order to verify the effectiveness and superiority of the PMD-YOLO proposed in this paper, the PMD-YOLO is used for experimental research on the constructed dataset of the pointer meter, and the target detection algorithms such as Faster region convolutional neural network (RCNN), YOLOv4, YOLOv4-Tiny, and YOLOv5-s are compared under the same conditions. The experimental results show that the mean average precision of the PMD-YOLO is 97.82%, which is significantly higher than the above algorithms. The weight of the PMD-YOLO is 9.38 M, which is significantly lower than the above algorithms. Therefore, the PMD-YOLO not only has high detection accuracy, but can also reduce the weight of the model and can meet the requirements of practical applications.

Keywords: pointer meter; object detection; YOLOv4-Tiny; GhostNet; attention mechanism

check for updates

Citation: Xu, W.; Wang, W.; Ren, J.; Cai, C.; Xue, Y. A Novel Object Detection Method of Pointer Meter Based on Improved YOLOv4-Tiny. *Appl. Sci.* **2023**, *13*, 3822. https://doi.org/10.3390/app13063822

Academic Editor: Luis Javier Garcia Villalba

Received: 22 February 2023
Revised: 9 March 2023
Accepted: 15 March 2023
Published: 16 March 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Pointer meters have been widely used in industrial places such as oil wells, oil fields, and substations because of their simple structure, low cost, long life, and strong anti-interference properties [1]. However, pointer meters generally do not have the functions of data transmission and data storage, and cannot realize automatic data collection and real-time transmission. The most traditional method is manual reading, but this method has the disadvantages of low efficiency, large error, and low timeliness [2], and when there is an unfavorable environment for manual reading outdoors, there is a possibility of safety accidents. With the development of the era, mobile robots and computer technology have been continuously developed and applied. At present, in the high-temperature, complex electromagnetic interference, corrosion, and other complex real environments, inspection robots with algorithm models have gradually replaced the manual meter detection, reading, early warning, and other applications, which can significantly reduce labor costs, improve the efficiency of instrumentation information conveyance, and ensure the safety and stability of industrial sites.

The detection of pointer meters is very important work and has been widely studied and applied. Zhang et al. [3] first used the localized pixel inhomogeneity factor model to improve the saliency of the pointer meter region in the binarized image, and then used Hough circle detection to detect the dial range of the meter, but the recognition results obtained by using Hough circle detections are often misread and miss detection due to light, shooting angle, complex background, etc.; in addition, different-sized circles need to set different parameters during detection, so the stability is poor. Li et al. [4] proposed a circle detection algorithm based on the line segments of circular bracket components to detect the meter and obtain circle center coordinates and radius of the meter. Xu et al. [5] used perspective transform detection and matching features to obtain various meter images. Mai et al. [6] used scale-invariant feature transform features to match the image to be detected with the template and used random sample consensus [7] to remove the wrong feature points, and thus the meter was detected. Since these methods require complex calculations and specific attributes of the meter, as well as a large number of template images of various types of meter being used to match features, these methods have the disadvantage of poor real-time performance and are easily limited in the actual complex environment, and the performance of automatic recognition needs to be improved. In order to accurately detect and locate meters in complex real-world scenes, it is necessary to have an object detection algorithm that can perform with high accuracy in a complex background environment. Image recognition and object detection through deep learning can effectively compensate for the shortcomings of traditional methods such as low accuracy, missed detection, and poor real-time performance, and can have better recognition effects on images with noise, occlusion, blur, and other interferences.

At present, image object detection methods based on deep learning have shown excellent performance in many fields; these object detection methods can be divided into two categories: two-stage method and one-stage method. The two-stage method includes region convolutional neural network (RCNN) [8], Faster RCNN [9], Mask RCNN [10], etc.; the steps of these algorithms can be divided into extracting object regions and using convolutional neural networks (CNNs) to classify and recognize regions, so it is called two-stage target detection algorithm. The two-stage method has slow speed and is unable to meet the target detection requirements in real-time environments, so the application of this method is limited. The one-stage method is used to view the object detection problem as a regression problem, and directly generates the class probability and location coordinate of the object without the region suggestion stage, so the final detection results can be obtained by a single detection. One-stage object detection methods are represented by single-shot detector (SSD) [11], Retina-Net [12], EfficientNet [13], and YOLO series algorithms [14–17], which have shown their advantages on both VOC datasets [18] and COCO datasets [19]. The one-stage object detection algorithm is not as accurate as the two-stage object detection algorithm, but the detection speed is better and it can accomplish the real-time detection of the object, so in the application of pointer meter detection, the one-stage detection method is generally used. Lin et al. [20] used the histogram normalized transformation algorithm to enhance the brightness and contrast of the image, and then used YOLOv3 to detect and obtain the position of the meter. Zhou et al. [21] used YOLOv5 to detect the meter and cropped the detected meter, and then used mathematical methods to read the pointers, scales, and meter centers of the cropped meter by secondary detection. Yang et al. [22] proposed an improved algorithm based on YOLOv3 to change the feature pyramid network (FPN) structure from the original unidirectional fusion to bidirectional fusion, and introduced Smooth-EIoU Loss to improve the algorithm's small object detection accuracy. Zhou et al. [23] proposed an improved algorithm named YOLO-SASE (self-adaption squeeze-and-excitation) based on the YOLO framework and super-resolution generative adversarial network, which improved the detection performance of small infrared objects in complex backgrounds by combining the SASE module, SPP module, and multilevel receptive field. Chen et al. [24] proposed a lightweight network based on the improved YOLOv4; they used PP-LCNet and depthwise separable convolution

to reduce model parameters, added an attention mechanism in the output layer of the backbone network to optimize the feature fusion structure, and used the SIoU loss function to improve the accuracy. Considering the applications in practical scenarios, the weight generated by the one-stage detection algorithms is too large, so they are not convenient to deploy in mobile devices with limited resources.

At present, equipping CNN models on mobile and embedded devices remains a considerable challenge due to limitations in storage space and computational speed, so lightweight networks have become a research topic in computer vision in recent years. The purpose of research on lightweight models is to minimize the number of parameters and complexity of the models while maintaining network detection accuracy, so that network models can be deployed on devices with limited storage space and computing power. Currently, there are more and more lightweight networks for this challenge on mobile and embedded devices, such as MobileNetv1-v3 [25–27] and EfficientNet [13] proposed by Google, ShuffleNet [28,29] and SqueezeNet [30] proposed by KuangShi Technology, and GhostNet [31] proposed by Huawei. These lightweight networks can reduce the parameters of convolutional neural networks by designing different convolution structures. In 2020, Alexey proposed a lightweight network based on YOLOv4 named YOLOv4-Tiny; its parameter is only 1/10 of YOLOv4. It can not only train the model with limited device resources, but also deploy the model in resource-constrained mobile devices to perform practical tasks related to object detection; however, the detection accuracy needs to be improved due to the reduced number of parameters of the model.

In order to make the model easy to deploy on mobile or embedded devices as well as to have higher accuracy, this paper designed a lightweight object detection network named pointer meter detection-YOLO (PMD-YOLO). PMD-YOLO takes YOLOv4-Tiny as a benchmark and further optimizes the network structure by combining with the lightweight network GhostNet. The overall contribution of this paper is summarized as follows.

- Based on YOLOv4-Tiny, a lightweight network GhostNet was used for feature extraction. This operation significantly reduced the weight of the model and greatly reduced the space required for network deployment.
- Bidirectional feature fusion and depthwise separable convolution were incorporated into the Neck stage; the feature layers of different sizes were further fused. This operation can make full use of the deep semantic features and shallow spatial location features of the pointer meter, thus improving the model detection performance.
- The improved receptive field block (RFB) was introduced behind the feature extraction network to enhance the network's attention to meter features by using dilated convolution that can simulate human visual mechanisms.
- A new channel attention mechanism was introduced in GhostNet, and two convolutional block attention modules (CBAMs) were introduced in the Neck stage to improve the network's attention to small- and medium-sized meters, so the object detection algorithm has improved the detection performance for small- and medium-sized objects.

2. PMD-YOLO

2.1. Framework of the PMD-YOLO

YOLO series are very efficient one-stage object detection models. In order to further improve the detection rate and accuracy of pointer meters, this paper proposes a new model named PMD-YOLO based on the improvement of YOLOv4-Tiny. The overall network structure of PMD-YOLO can be seen Figure 1. It can be seen from the figure that PMD-YOLO can be divided into three parts; they are Backbone, Neck, and YOLO Head. In Backbone, GhostNet was introduced to extract the feature information of the input images, which significantly reduced the model volume and the number of computational parameters. YOLO Head uses the extracted features for category and confidence prediction for the final output; Neck is located between the Backbone and YOLO Head, and it can further enhance the diversity and robustness of features by introducing the improved RFB module, CBAM, and improved FPN.

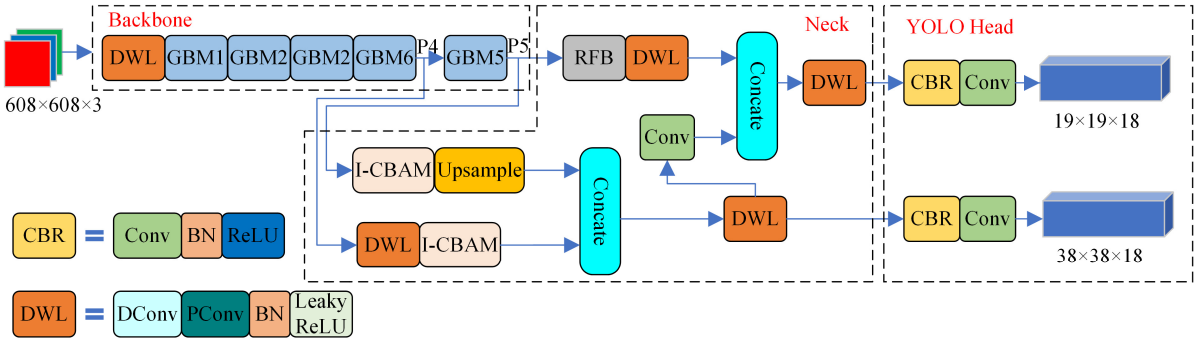


Figure 1. The structure of PMD-YOLO.

As the main feature extraction part of the model, the network structure of Backbone is generally more complex, thus resulting in a larger model size. In order to achieve a better lightweight model, GhostNet was chosen as the Backbone of the model by comparing with other lightweight networks. In GhostNet, the Ghost Module uses 3 × 3 depthwise separable convolution combined with 1 × 1 convolution to replace the original 1 × 1 convolution. In order to further reduce the number of model parameters, CBAM with small number of parameters was used to replace the original channel SE attention module. The CBAM is detailed in Section 2.3.

It can be seen from the overall structure of PMD-YOLO that the output of the Backbone includes two scale feature maps, with the size of 19 × 19 (feature layer P5), 38 × 38 (feature layer P4). In Neck, firstly, an improved RFB module was used to increase the receptive field of the P5, and two CBAMs were added between feature fusion and depthwise separable convolution. Then, the idea of bidirectional feature fusion was used to further refine and fuse the output information of the backbone, so as to improve the detection performance of the PMD-YOLO model. The network features were enhanced and fused, and two feature maps with scale 19 × 19 and 38 × 38 were obtained, respectively. Each feature map was outputted by YOLO Head as a tensor with a channel dimension of 18, i.e., 3 × (4 + 1 + 1), where "3" denotes three different scales of anchor frames, "4" denotes the center coordinates (x,y), width (w) and height (w) of each anchor frame, the first "1" represents the confidence of the prediction result, and the second "1" represents the object category number. Since this article focuses on the detection task of circular pointer meters, only one target category is set, which is simply marked as "meter". Finally, CIoU was used to calculate the loss of the prediction frame and the true frame, and the best-scoring prediction frame was filtered by using nonmaximal suppression (NMS). It should be noted that CIoU is a metric for measuring the distance between two bounding boxes, which can be applied to compute the distance between predicted bounding boxes and ground truth bounding boxes in object detection.

2.2. Ghost Backbone Network

(1) Ghost Module

Han [31] found that combining a small number of traditional convolution results with the results of linear transformation can maintain the network accuracy while reducing the number of parameters of the model. A comparison between the traditional convolution process and the Ghost Module convolution process is shown in Figure 2. Compared with the traditional convolution, the Ghost Module convolution process is divided into three steps. Firstly, the 1 × 1 traditional convolution is used to obtain a feature map with fewer channels, then the feature map generated in the first step is used to obtain more feature maps by using the cheap operation, and, finally, the feature maps obtained in the first two steps are concatenated together to form a new feature map as the output feature map of Ghost Module. Here, the cheap operation is a 3 × 3 deep separable convolution.

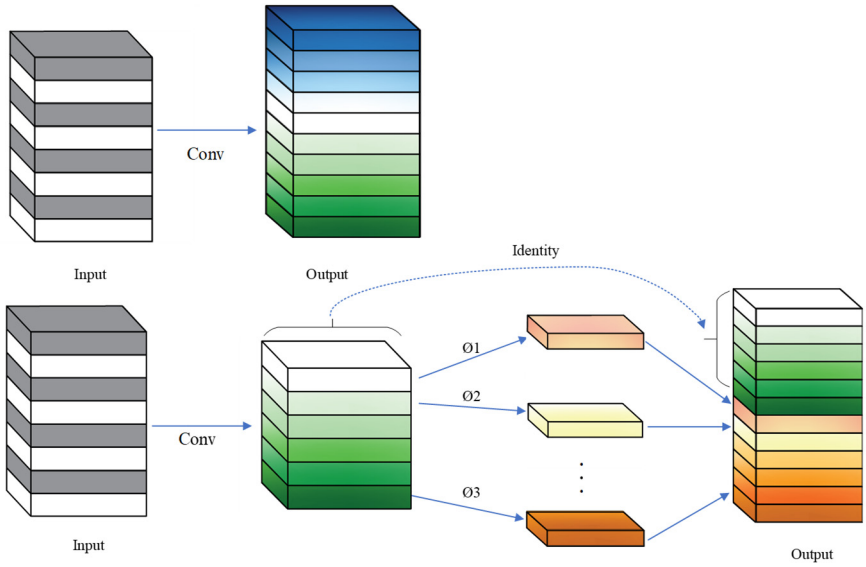


Figure 2. Traditional convolution and Ghost Module convolution. The green and blue layers represent the feature maps obtained by traditional convolution, and the yellow layers represent the feature maps obtained by cheap operation.

(2) Ghost Bottleneck Module

The Ghost Bottleneck Module (GBM) adopts the same residual structure as ResNet to solve the problem that the gradient of the shallow network disappears during network training; the structure of Ghost Bottleneck is shown in Figure 3. It can be seen from the figure that the network structure containing the Ghost Module is the backbone, and the other forward route is the residual. The Ghost Bottleneck Module with stride = 2 adds a depthwise separable convolution with stride = 2 between the two Ghost Modules to compress the width and height of the feature layer. A few Ghost Bottleneck Modules add channel attention modules before the second Ghost Module, and these channel attention modules increase or decrease the weight occupied by the features according to their importance. In the original GhostNet, the authors used the SE channel attention mechanism, and this paper used CBAM with a smaller number of parameters, described in Section 2.3.

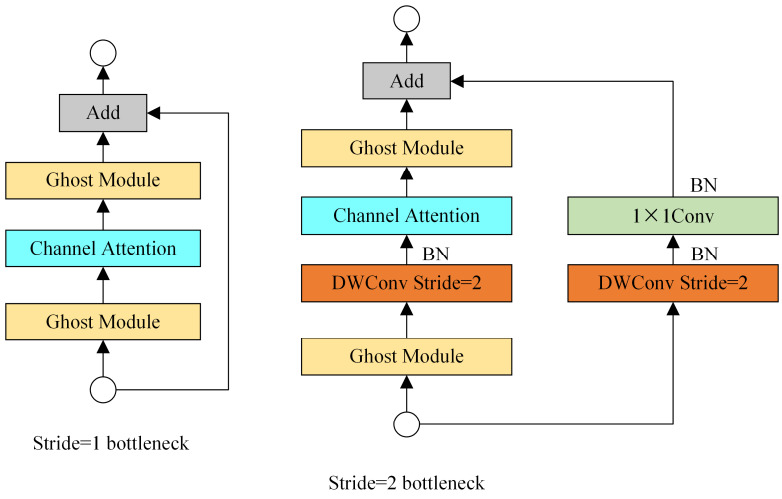


Figure 3. The structure of Ghost Bottleneck.

(3) GhostNet

The whole of GhostNet consists of a depthwise separable convolutional block and multiple GBMs, which can be divided into five stages according to the number of GBM.

Each stage is denoted by "GBMx". "x" indicates the number of Ghost Bottleneck Modules, and the output size of the feature map of each stage is shown in Figure 4. The channel attention mechanism is mixed and interspersed in each Ghost Bottleneck Module to extract more important features. The final output is feature layers P4 ($38 \times 38 \times 112$) and P5 ($19 \times 19 \times 160$) with different size, and these two feature maps are fed into the feature fusion stage.

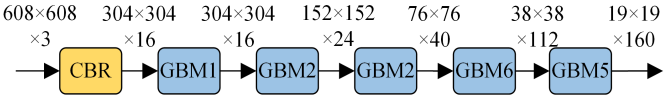


Figure 4. The structure of GhostNet.

2.3. Attentional Mechanism

The attention mechanism can increase the representativeness of the network by extracting important features and suppressing unnecessary features, while global maximum pooling and global average pooling of feature channels can extract important features and suppress unnecessary features. To further improve the network performance, this paper introduces CBAM with a small number of parameters in the network, whose structure is shown in Figure 5. It consists of a channel attention mechanism and a spatial attention mechanism, where the channel attention mechanism makes the network focus on the "what" of the object features and the spatial attention mechanism makes the network focus on the "where" of the object features.

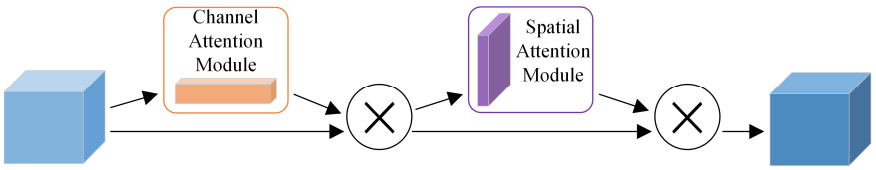


Figure 5. The structure of CBAM.

(1) Channel attention mechanism

The structure of the channel attention mechanism is shown in Figure 6, and it can be seen from the figure that the input feature map is F(W $\times$ H $\times$ C). The basic process of the channel attention mechanism is as follows. First, global maximum pooling and global average pooling are performed on the height and width of the input feature map to generate two feature layers (F_{max}, F_{avg}) with size $1 \times 1 \times$ C to map the channel information, and two 1×1 convolution kernels are used to adjust the number of channels of F_{max} and F_{avg} to integrate the feature information; ReLU activation function is used once between the two 1×1 convolutions, and the adjusted two feature layers are added element by element. Then, the channel attention feature map M_c is obtained through the operation of the sigmoid activation function; finally, F_c can be obtained by multiplying the feature map M_c and the input feature map F element by element. In experiments, the model performance is improved and the number of model parameters is reduced by using this channel attention mechanism instead of the SE mechanism in GhostNet. The calculation formula of the whole process is as follows.

$$M_c = \sigma(f_C^{1\times1}(\Phi(f_{C/8}^{1\times1}(MaxPool(F))) + f_C^{1\times1}(\Phi(f_{C/8}^{1\times1}AvgPool(F))))) \quad (1)$$

$$F_c = M_c \otimes F \quad (2)$$

where σ denotes sigmoid activation function, Φ denotes ReLU activation function, MaxPool and AvgPool denote global max pooling and global average pooling, respectively, $f_{C/8}^{1\times1}$ denotes convolution kernel with size 1×1 and output channel C/8, $f_C^{1\times1}$ denotes convolution kernel with size 1×1 and output channel C, and $\otimes$ denotes element-by-element multiplication.

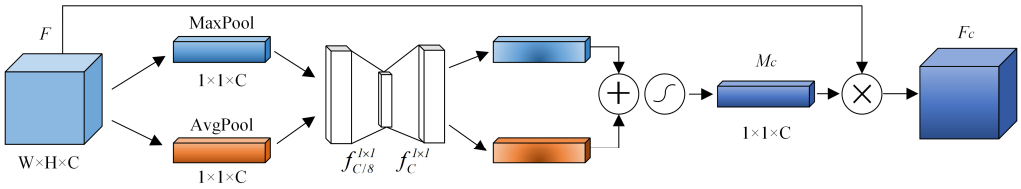


Figure 6. Channel attention mechanism.

(2) Spatial attention mechanism

The spatial attention mechanism is shown in Figure 7. The F_c(W × H × C) generated by the previous channel attention mechanism is used as the input. The basic process of the spatial attention mechanism is as follows. First, global average pooling and global max pooling are performed in the channel dimensions to generate two feature maps with size W × H × 1, and two feature maps are concatenated together in the channel dimensions to obtain a feature map with size W × H × 2. Then, the number of channels of the feature map is adjusted by using a convolution kernel of size 7 × 7, and the spatial attention feature map M_s is obtained by the operation of the sigmoid activation function; finally, F_s can be obtained by multiplying the feature map M_s and the input feature map F_c element by element. The calculation formula of the whole process is as follows.

$$M_s = \sigma(BN(f^{7\times7}([AvgPool(F_c); MaxPool(F_c)]))) \quad (3)$$

$$F_s = M_s \otimes F_c \quad (4)$$

where σ denotes the sigmoid activation function, MaxPool and AvgPool denote global max pooling and global average pooling, respectively, $f^{7\times7}$ denotes convolution kernel with size 7 × 7, BN denotes batch normalization, and ⊗ denotes element-by-element multiplication.

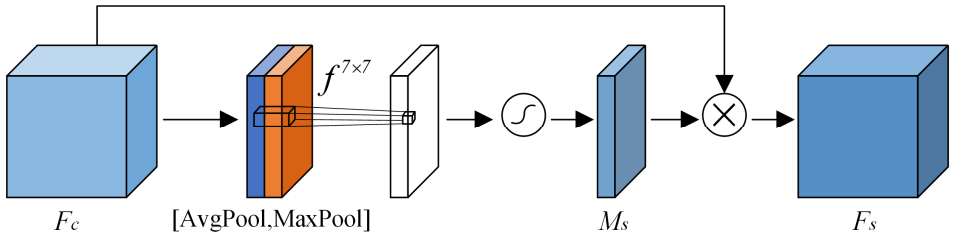


Figure 7. Spatial attention mechanism.

2.4. Enhanced Receptive Field Block

The receptive field mechanism of the human visual will focus on the center of the object, and the feature attention within the receptive field gradually decreases from the center to the surroundings. The RFB module is a method used to simulate the receptive field mechanism of the human visual. It can increase the receptive field, strengthen the importance of center features, reduce the attention of edge features, and improve the ability of the network to extract effective features.

The RFB module uses the dilated convolution and controls the size of the receptive field by using the dilation rate. Since a too-large receptive field can introduce too much background environment, in order to enhance the network's detection of small- and medium-sized objects, this paper set the maximum of the dilation rate to 3 and replaced the original 3 × 3 convolution kernel with 1 × 3 and 3 × 1 convolution kernels to achieve the effect of parameter reduction. The improved RFB module is shown in Figure 8, and it is placed behind the output feature layer P5 to enhance the detection capability of small- and medium-sized object objects.

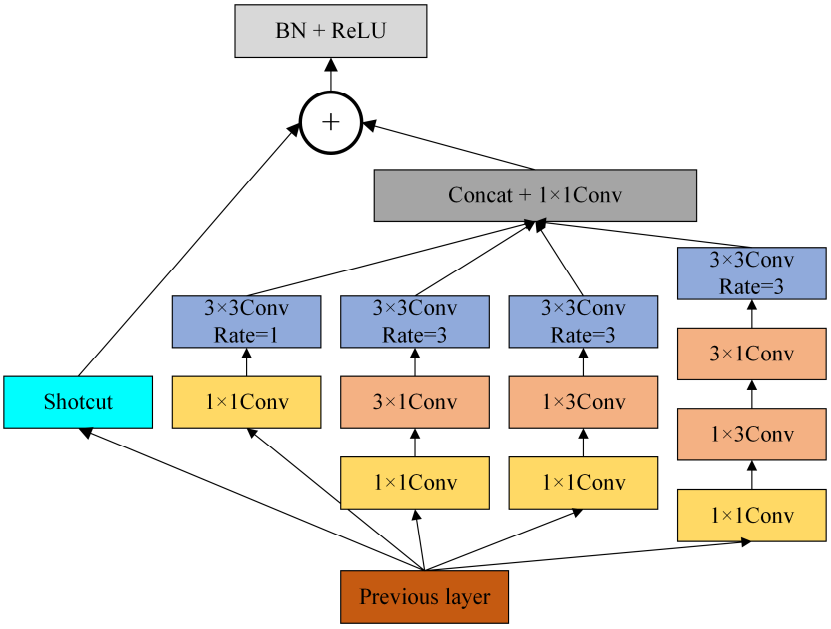


Figure 8. The structure of the improved RFB.

2.5. Bidirectional Feature Fusion

It is well known that in CNN networks, the higher-level features contain richer semantic information and the bottom-level features contain richer spatial location information. In YOLOv4-Tiny, the feature fusion stage uses a bottom-up FPN, and this structure will have the problem that the shallow information is easily lost during the transfer process. In order to make full use of the spatial location information and semantic information contained in the multilevel feature map, we optimized the feature fusion method.

As shown in Figure 9, we constructed a top-down and bottom-up bidirectional fusion structure based on the traditional FPN structure, and added two CBAMs to the structure. Depthwise separable convolutional blocks were used to integrate the feature maps before and after feature fusion, and two feature maps were obtained in the feature fusion stage by following the two YOLO Heads of YOLOv4-Tiny for detection. The fused feature maps have richer underlying spatial location information and higher-level semantic information.

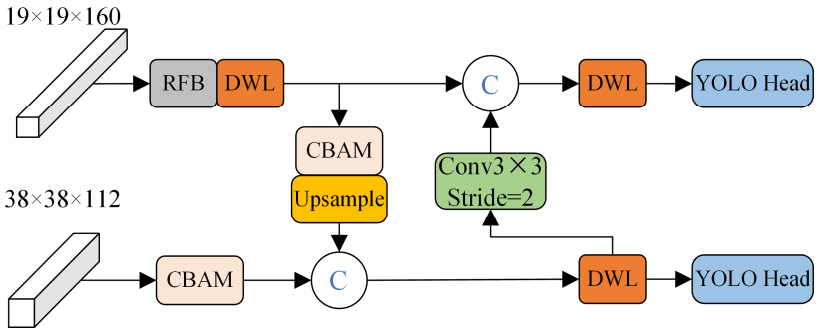


Figure 9. The structure of the improved FPN.

3. Experiments and Discussions

In order to verify the detection capability of PMD-YOLO for a pointer meter in a real complex background environment, this section first presents the experimental preparation, including the dataset preparation and evaluation methods, then the ablation experiment of PMD-YOLO is carried out, and the comparison experiment with the existing excellent target detection algorithm is carried out under the same conditions.

3.1. Experimental Environment and Settings

(1) Description of the Dataset

Due to the lack of public datasets for pointer meters, pointer meter images were collected from the web by using web crawlers and from real scenes by using mobile phones, so a high-standard dataset was created. Firstly, web images were scraped by using "pointer meter" and "pointer gauge" as keywords. However, since these images varied in quality and had duplicates, the repetitive and poor-quality images were filtered out manually. The remaining images were still insufficient for deep learning purposes, so they were supplemented by the photos of pointer meters taken by mobile phones in realistic settings. Then, the two sources of images were mixed and the dataset images were labeled as "meter" by using LabelImg software to generate YOLO format labels. Finally, some images were randomly cropped, dynamic fuzzy data were enhanced, and a dataset containing 2000 pointer meter images with 2575 pointer meter targets was obtained. In addition, all the data were randomly sampled and divided into training set, validation set, and test set at a ratio of 8:1:1.

(2) Experimental parameters and environment

To ensure the fairness of the experiment, the initial training parameters of each group of experiments were set to be the same, the size of all input images was adjusted to $608 \times 608 \times 3$, each training batch was set to 16, the num_works was set to 2, all models were trained for 200 epochs, the initial learning rate was set to 0.01 and CosineAnnealing was used to reduce the learning rate, the minimum learning rate was set to 0.0001, SGD was selected as the optimizer, the mosaic data enhancement method was enabled for the first 70% of epochs, the confidence threshold was set to 0.5, and the IoU threshold was set to 0.3. The computational resources required for the experiment are shown in Table 1.

Table 1. The computational resources required for the experiment.

Component	Versions, Model Number or Size
CPU	Intel(R) Core(TM) i7-12700H
GPU	NVIDIA RTX 3060
VRAM	6 GB
RAM	16 GB
OS	Windows 11
CUDA	v11.6
Pytorch	V1.13.0
cuDNN	V8302
Python	V3.9

3.2. Model Evaluation Indexes

In this paper, precision (P), recall (R), $F1$, and mean average precision (mAP) were used as the evaluation indexes of network model accuracy, and the calculation formulas are as follows.

$$P = \frac{TP}{TP + FP} \times 100\% \tag{5}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{6}$$

$$F1 = \frac{2PR}{P + R} \tag{7}$$

$$mAP = AP = \int_0^1 P(R)dR \times 100\% \tag{8}$$

In the equations, P denotes the proportion of positive samples among all samples; R denotes the proportion of samples predicted to be positive among all positive samples; TP is the number of prediction frames correctly matched with labeled frames; FP is the

number of prediction frames predicted incorrectly; *FN* is the number of labeled frames not predicted; *F*1 is the summed average of precision and recall; *AP* denotes the average precision of a category, and *mAP* indicates the mean value of all categories of *AP*, which is usually used as a comprehensive index for accuracy evaluation. The object to be detected in this paper is only one category of pointer meter, so *mAP* is equal to *AP*. In addition to the previously described indexes, the size of the network model and the number of calculation parameters of the network model were also introduced as evaluation indexes.

3.3. Experimental Results

(1) Ablation experiments

In order to verify the impact of each module in the PDM-YOLO network on the overall detection performance compared to the YOLOv4-Tiny baseline network, ablation experiments were conducted on PDM-YOLO by using "Model X" instead of the network name with specific network module. For example, YOLOv4-Tiny is temporarily named as Model 1. The experimental results are shown in Table 2. In the table, "✓" indicates the corresponding method used; GhostNet(CBAM) indicates that the original SE attention mechanism in GhostNet is replaced by the channel attention mechanism in CBAM.

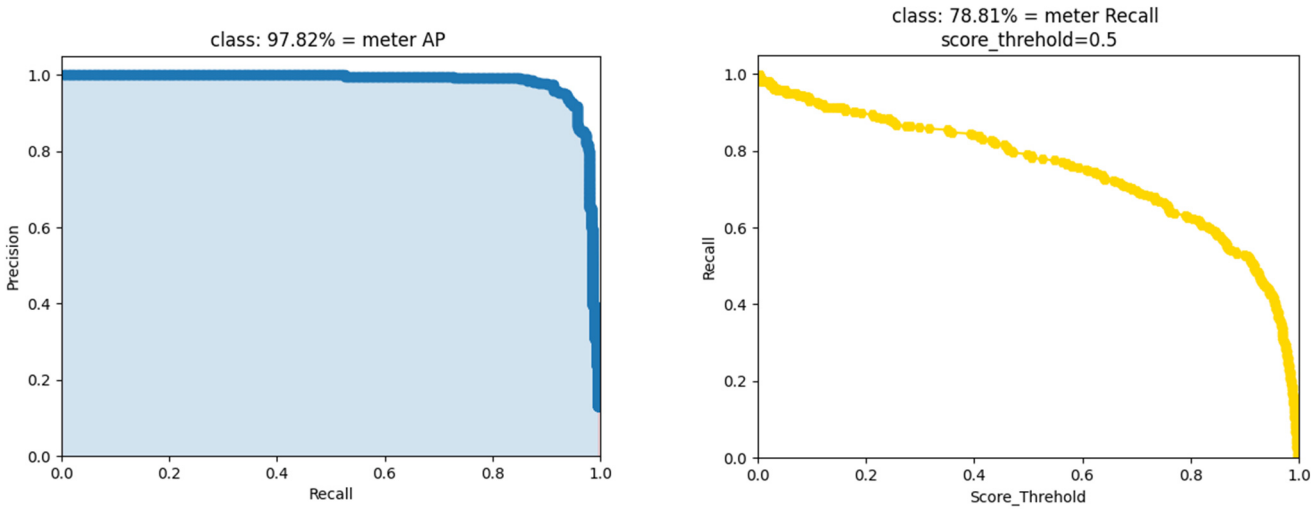
Table 2. Ablation experiments of PMD-YOLO.

No.	YOLOv4-Tiny (Base)	GhostNet (SE)	GhostNet (CBAM)	Improved RFB	Improved Fusion	CBAM	Recall Rate (%)	*mAP* @0.5 (%)	Weight Size/(MB)
1	✓						72.19	92.64	22.4
2	✓	✓					76.82	93.95	10.9
3	✓		✓				76.16	94.32	8.75
4	✓		✓	✓			77.81	95.51	8.68
5	✓		✓	✓	✓		78.53	96.16	9.18
6	✓		✓	✓	✓	✓	78.81	97.82	9.38

From the results in Table 2, it can be seen that the GhostNet lightweight feature extraction network replaces the backbone network in YOLOv4-Tiny; it reduced the model size by 51.3%, and improved the recall rate and mAP by 4.63% and 1.31%, respectively. When the channel attention mechanism in CBAM is used to replace the SE channel attention mechanism, the features are extracted through global average pooling and global maximum pooling; although the recall rate is slightly reduced, the *mAP* is increased by 0.37%, and the model size is reduced by 60.93% compared with the YOLOv4-Tiny network. When the improved RFB module is added to the model, the importance of the central feature of the receptive field of the model is increased; not only are the recall rate and *mAP* increased by 1.65% and 1.19%, respectively, but the size of the model is also further reduced. The size of the model is 61.25% of the YOLOv4-Tiny network model, with the highest lightness. When the improved feature fusion is incorporated into the model, the fusion of feature maps with different scales is enhanced, the utilization rate of underlying features is improved, and the recall rate and *mAP* are increased by 0.72% and 0.65%, respectively. When the CBAM attention mechanism is introduced into the model, the recall rate and *mAP* can reach 78.81% and 97.82%, respectively, by increasing the attention to important features and restraining the attention to unnecessary features, so that the PMD-YOLO network can accurately detect the target of the pointer meter. The experimental results show that the improvement of YOLOv4-Tiny in this paper is effective; these improvements not only improve the detection accuracy of the model, but also reduce the size of the model, thus making the model more practical.

Figure 10 shows the precision–recall (P–R) curve and recall curve generated for the PDM-YOLO with input size of 608 × 608 on the constructed dataset of the pointer meter. In the P–R curve, the area below the curve is the value of *AP*, and it can be seen from the figure that the average accuracy of the pointer meter is 97.82%. The recall curve shows the

proportion of the number of correctly predicted target samples to the total positive samples under the condition that the confidence threshold is 0.5; it can be seen from the figure that recall is equal to 78.81%. However, under the same conditions, the *mAP* of YOLOv4-Tiny is 92.64%, and the recall is 72.19%. These results confirm the effectiveness and superiority of the method proposed in this paper.



(**a**) Precision–recall (**b**) Recall

Figure 10. Curves of precision–recall and recall.

(2) Comparison experiments with other models

In order to verify the superiority of PMD-YOLO, we selected several target detection algorithms and compared their performances with that of PMD-YOLO under the same dataset. These algorithms include the classical two-stage object detection algorithm (Faster RCNN), one-stage object detection algorithm (YOLOv4) and its related improved algorithms, and YOLOv5-s with small volume. The results of the comparison experiments are shown in Table 3.

Table 3. Performance comparison of PMD-YOLO and other detection algorithms.

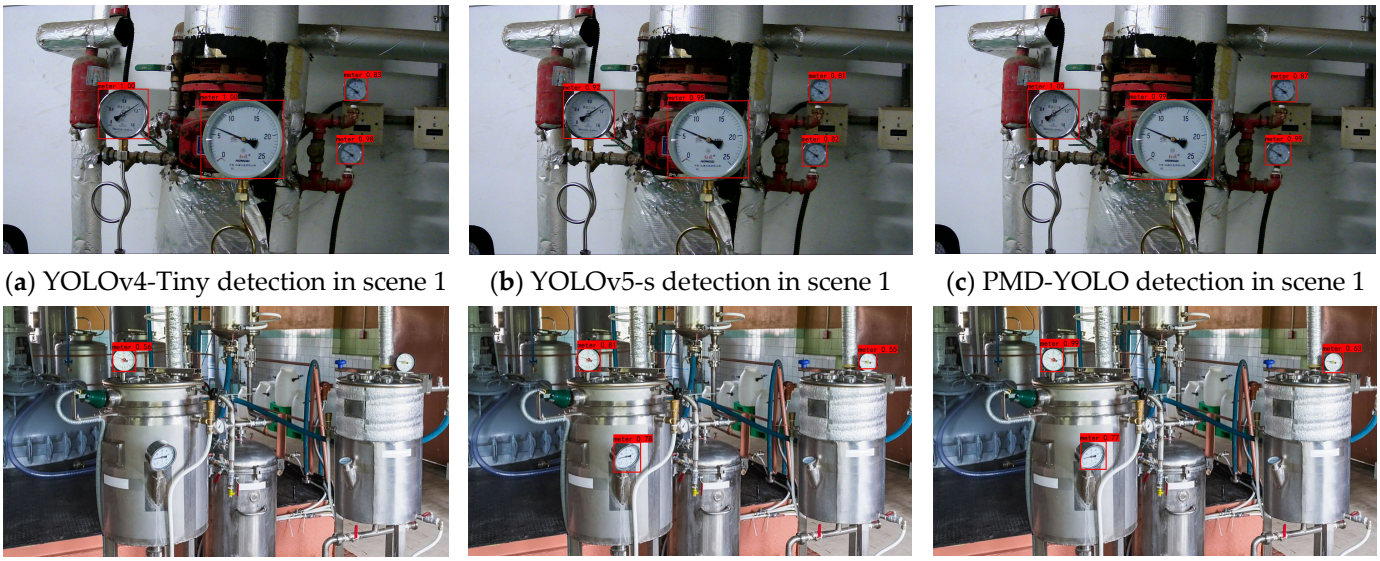
Model	Backbone	Precision (%)	Recall (%)	*F*1	*mAP* (%)	Weight Size (MB)	Parameter
Faster RCNN	Resnet50	84.86	98.34	0.91	98.31	102	136×10^6
YOLOv4	CSPDarknet53	93.53	73.85	0.85	93.42	244	63.9×10^6
YOLOv4-MobileV1	MobileNetV1	98.12	79.42	0.88	94.67	51.1	12.3×10^6
YOLOv4-MobileV2	MobileNetV2	96.64	79.75	0.88	94.27	46.5	10.4×10^6
YOLOv4-MobileV3	MobileNetV3	98.86	80.42	0.89	95.62	53.7	11.3×10^6
YOLOv4-Tiny	CSPDarknet53-tiny	98.64	72.19	0.83	92.64	22.4	5.9×10^6
YOLOv5-s	CSPDarknet	96.06	80.74	0.90	96.72	27.1	7.1×10^6
PMD-YOLO	I-GhostNet	99.17	78.81	0.88	97.82	9.38	3.1×10^6

It can be seen from Table 3 that YOLOv4-MobileV3 has the best overall performance among network models whose backbone network is replaced by the MobileNet series networks; therefore, YOLOv4-MobileV3 represents the YOLOv4 model whose backbone is MobileNet series networks. From the data in Table 3, it can be seen that although the recall rate and *F*1 of PMD-YOLO are slightly lower than other individual algorithms, PMD-YOLO has obvious advantages in detection accuracy, number of computed parameters, and weight size. The precision of PMD-YOLO is 99.17%, which is 5.64%, 0.31%, 0.53%, and 3.11% higher than that of YOLOv4, YOLOv4-MobileV3, YOLOv4-Tiny, and YOLOv5-s, respectively; the *mAP* of PMD-YOLO is 97.82%, which is 4.4%, 2.2%, 5.18%, and 1.1% higher than that of

YOLOv4, YOLOv4-MobileV3, YOLOv4-Tiny, and YOLOv5-s, respectively. Meanwhile, the weight volume and calculation parameters of PMD-YOLO are also minimized; the weight volume of PMD-YOLO is reduced to 9.38 MB, which is 3.8%, 17.5%, 41.9%, and 34.6% of that of YOLOv4, YOLOv4-MobileV3, YOLOv4-Tiny, and YOLOv5-s, respectively; the computational parameters of PMD-YOLO are reduced to 3.1×10^6, which is 4.9%, 27.4%, 52.5%, and 43.7% of that of YOLOv4, YOLOv3, YOLOv4-Tiny, and YOLOv5-s, respectively. In addition, the two-stage object detection algorithm Faster RCNN is 0.49% higher than PMD-YOLO in terms of *mAP*, but the weight volume and the amount of computational parameters are too large to be applicable on mobile devices. In a comprehensive view, PMD-YOLO effectively improved the feature extraction ability of the algorithm by replacing the backbone network, introducing CBAM, improving the RFB module, and optimizing the feature fusion method, while reducing the computational parameters of YOLOv4-Tiny, and the object detection effect was significantly improved.

(3) Comparison of detection effect

In this paper, YOLOv4-Tiny, YOLOv5-s, and PMD-YOLO, which are also lightweight networks, were chosen as the experimental groups to compare the detection effect of pointer meters in real scenes, and Figure 11 shows the detection effect of the above three models for pointer meter in complex background scenes. In the actual scene 1 shown in Figure 11a–c, all three models do not produce any missed detection due to small objects, but the recognition rate of PMD-YOLO for each object is higher than that of the other two algorithms. In the actual scene 2 shown in Figure 11d–f, the meter detection results of YOLOv4-Tiny have serious missed detection, and the meter detection results of YOLOv5-s and PMD-YOLO also have missed detection, but such too-small objects lose their reading significance because the too-small meters are not able to detect other features required for meter reading, and PMD-YOLO is higher than YOLOv5-s in terms of accuracy. It can be seen from the experimental comparison results that the detection results of PMD-YOLO are closer to the minimum external rectangle of the meter target, which indicates that PMD-YOLO can detect the position of the meter more accurately.



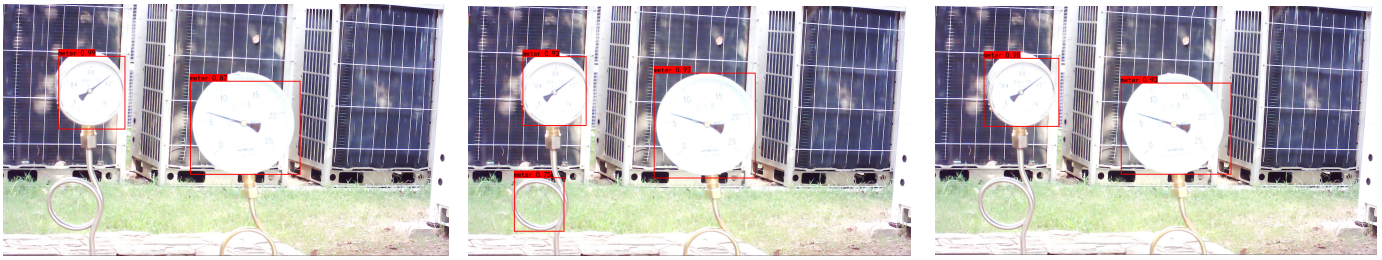
(**a**) YOLOv4-Tiny detection in scene 1 (**b**) YOLOv5-s detection in scene 1 (**c**) PMD-YOLO detection in scene 1

(**d**) YOLOv4-Tiny detection in scene 2 (**e**) YOLOv5-s detection in scene 2 (**f**) PMD-YOLO detection in scene 2

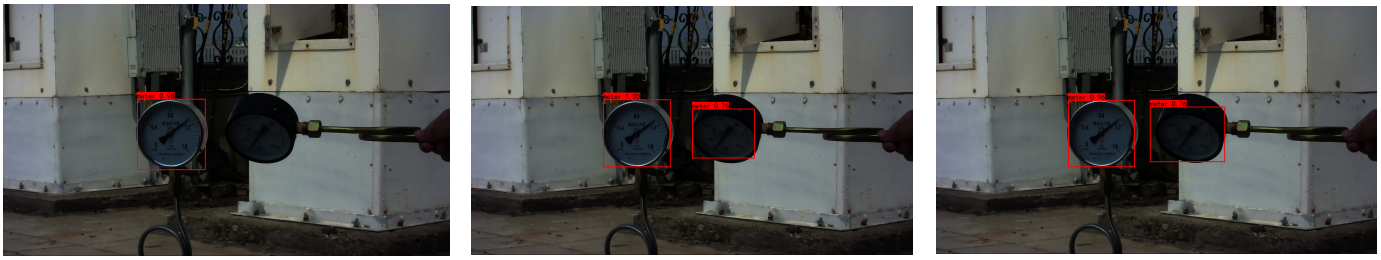
Figure 11. Comparison detection results of three models on pointer meters in different complex background scenes. The object confidence scores in each graph are shown from left to right: (**a**) shows 1.00, 1.00, 0.98, and 0.83; (**b**) shows 0.92, 0.95, 0.82, and 0.81; (**c**) shows 1.00, 0.99, 0.99, and 0.87; (**d**) shows 0.56; (**e**) shows 0.81, 0.78, and 0.55; (**f**) shows 0.99, 0.77, and 0.63.

Figure 12 shows the detection results of three models under overexposure, underexposure, and oblique angle shooting conditions. Under overexposed environments,

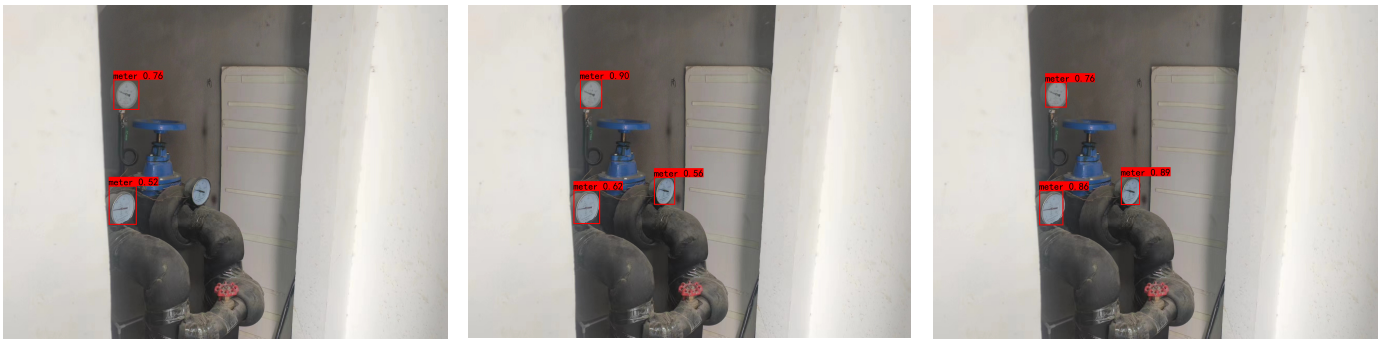
YOLOv5-s's bounding box was unaffected, but there was one false positive. The other two algorithms showed a slight offset of the bounding box. Under underexposed environments, YOLOv4-Tiny missed one target. YOLOv5-s and PMD-YOLO were able to recognize the tilted meter, but their detection accuracies also suffered a minor impact. Under oblique angle shooting conditions, YOLOv4-Tiny missed one target again, and the detection accuracy of all three models was affected to varying degrees. However, PMD-YOLO maintained a relatively high level of recognition accuracy overall. In summary, YOLOv4-Tiny and YOLOv5-s, respectively, missed and falsely detected targets under various adverse conditions, indicating low robustness. PMD-YOLO also experienced slight deviations and decreases in prediction accuracy under various adverse conditions, but it can be seen that PMD-YOLO has good overall detection performance.



(**a**) YOLOv4-Tiny detection in case 1 (**b**) YOLOv5-s detection in case 1 (**c**) PMD-YOLO detection in case 1



(**d**) YOLOv4-Tiny detection in case 2 (**e**) YOLOv5-s detection in case 2 (**f**) PMD-YOLO detection in case 2



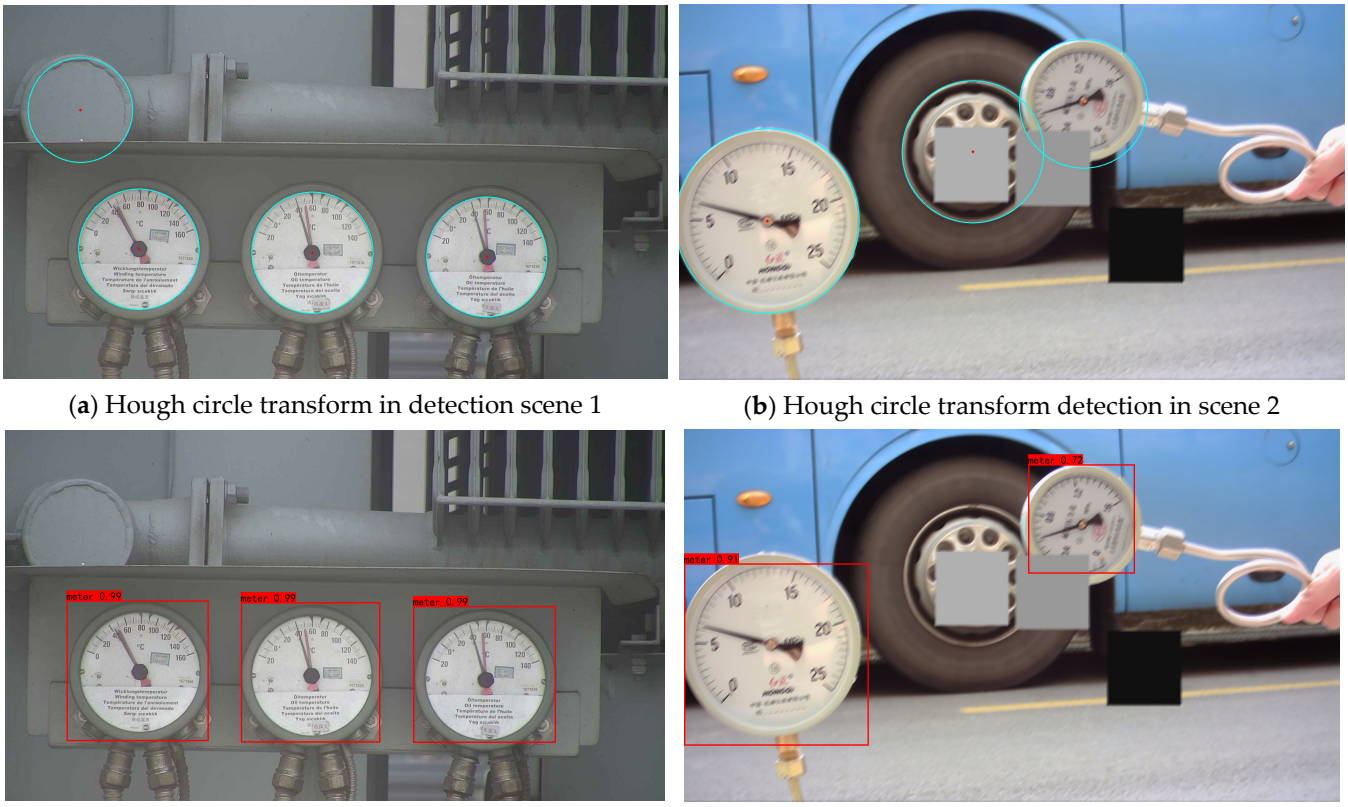
(**g**) YOLOv4-Tiny detection in case 3 (**h**) YOLOv5-s detection in case 3 (**i**) PMD-YOLO detection in case 3

Figure 12. Comparison detection results of three models on pointer meter in different conditions; (**a**–**c**) are the results of overexposure shooting; (**d**–**f**) are the results of underexposed shooting; (**g**–**i**) are the results of oblique angle shooting. The object confidence scores in each graph are shown from left to right: (**a**) shows 0.99 and 0.87; (**b**) shows 0.75, 0.93, and 0.93; (**c**) shows 0.98 and 0.93; (**d**) shows 0.98; (**e**) shows 1.00 and 0.76; (**f**) shows 0.96 and 0.76; (**g**) shows 0.52 and 0.76; (**h**) shows 0.62, 0.90, and 0.56; (**i**) shows 0.86, 0.76, and 0.89.

(4) Comparison with traditional methods

In order to show the advantages of the method proposed in this paper compared with traditional target detection methods, the comparative experiment of Hough circle transform and PMD-YOLO in different real scenes was carried out, and the experimental results are shown in Figure 13. In order to show that Hough circle transform is more easily

affected by interference information in environments with quasi-circular features, occlusion, and other interference factors, while PMD-YOLO has stronger anti-interference ability, the application scene completely different from Figure 11 was selected during the experiment. It can be seen from Figure 13 that when Hough circle transform is used to detect the meter, it is very easily interfered with by other circular features in the picture, and when the meter part is partially occluded, the position deviation of the detected target will be large. In addition, during the experiment, it was found that when there are many meters of different sizes in the image, different parameters need to be set when Hough circle transform is used to detect the meter, and the detection stability is poor. However, when the PMD-YOLO algorithm is used to detect the meter in this case, these shortcomings do not exist; even if the meter is partially occluded, the meter position can still be accurately detected.



(**a**) Hough circle transform in detection scene 1

(**b**) Hough circle transform detection in scene 2

(**c**) PMD-YOLO detection in scene 1

(**d**) PMD-YOLO detection in scene 2

Figure 13. Comparison detection results of Hough circle transform and PMD-YOLO. The object confidence scores in (**c**,**d**) are shown from left to right: (**c**) shows 0.99, 0.99, and 0.99; (**d**) shows 0.91 and 0.72.

4. Conclusions

In this paper, a one-stage lightweight detection model that can be used for pointer meters was proposed for the purpose of optimizing the model weights while improving the detection performance of traditional object detection models in complex real-world environments. In terms of model weight quantification, a lightweight network named GhostNet and an improved RFB module were introduced to significantly reduce the number of network model parameters; in terms of detection performance improvement, a new channel attention mechanism was introduced in GhostNet, the feature fusion stage was improved to enhance the feature utilization, and two CBAMs were added in the feature fusion stage to increase the feature extraction capability for small- and medium-sized objects. In order to verify the effectiveness and superiority of the model proposed in this paper, an experimental study of the proposed method was carried out with the self-made pointer meter dataset as the research object, and a comparative experimental study was

carried out with other similar methods. The experimental results show that the overall performance of the proposed model has obvious advantages; the *mAP* of the model can reach 97.82%, and the weight of the model is only 9.38 M. This shows that the model proposed in this paper not only improves the detection accuracy of the pointer meter, but also reduces the size of the model. The model can be easily deployed on mobile devices to meet the needs of practical applications.

In experiments, it was also observed that PMD-YOLO's performance would decline when it detected too-small objects or objects taken in images with poor light and angle conditions. To overcome these limitations, these problems will be further studied in the future work to reduce the negative impact of these conditions on the model. Furthermore, since the model introduces depthwise separable convolution, there is a partial performance degradation in terms of detection speed, but it still has high real-time detection capability. In future work, the network structure can be further optimized to improve the detection speed of the model while reducing the model size and improving the detection accuracy.

Author Contributions: W.X.: Methodology, formal analysis, software, validation, and writing—original draft. W.W.: Funding acquisition, project administration, formal analysis, and validation. J.R.: Methodology, formal analysis, investigation, and writing—review and editing. C.C.: Funding acquisition, methodology, project administration, and writing—review and editing. Y.X.: Formal analysis, investigation, and writing—original draft. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Nature Science Foundation of Hebei Province grant no. E2020402060, Special Project for Transformation of Major Technological Achievements in Hebei Province grant no. 22293601Z, and Key Laboratory of Intelligent Industrial Equipment Technology of Hebei Province (Hebei University of Engineering) under grant no. 202206.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

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

References

1. Xu, L.; Fang, T.; Gao, X. An Automatic Recognition Method of Pointer Instrument Based on Improved Hough Transform. *Proc. SPIE* **2015**, *9675*, 96752T.
2. Li, Q.; Fang, Y.; He, Y.; Yang, F.; Li, Q. Automatic Reading System Based on Automatic Alignment Control for Pointer Meter. In Proceedings of the IECON 2014—40th Annual Conference of the IEEE Industrial Electronics Society, Dallas, TX, USA, 29 October–1 November 2014; pp. 3414–3418.
3. Zhang, L.; Fang, B.; Zhao, X.; Zhang, H.-S. Pointer-Type Meter Automatic Reading from Complex Environment Based on Visual Saliency. In Proceedings of the 2016 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), Jeju Island, Republic of Korea, 10–13 July 2016; pp. 264–269.
4. Li, D.; Li, W.; Yu, X.; Gao, Q.; Song, Y. Automatic Reading Algorithm of Substation Dial Gauges Based on Coordinate Positioning. *Appl. Sci.* **2021**, *11*, 6059. [CrossRef]
5. Xu, S.; Hou, C.; Wang, Z.; Li, B. Pointer Gauge Adaptive Reading Method Based on a Double Match. *Meas. Sci. Technol.* **2020**, *31*, 115002. [CrossRef]
6. Mai, X.; Li, W.; Huang, Y.; Yang, Y. An Automatic Meter Reading Method Based on One-Dimensional Measuring Curve Mapping. In Proceedings of the 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), Lanzhou, China, 24–27 August 2018; pp. 69–73.
7. Ruzgienė, B.; Förstner, W. Ransac for Outlier Detection. *Geod. Cartogr.* **2012**, *31*, 83–87. [CrossRef]
8. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
9. 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*, 1–9. [CrossRef] [PubMed]
10. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.

11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9905, pp. 21–37, ISBN 978-3-319-46447-3.
12. 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.
13. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.
14. 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), IEEE, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
15. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
16. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
17. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
18. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338.
19. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2014; Volume 8693, pp. 740–755, ISBN 978-3-319-10601-4.
20. Lin, Y.; Zhong, Q.; Sun, H. A Pointer Type Instrument Intelligent Reading System Design Based on Convolutional Neural Networks. *Front. Phys.* **2020**, *8*, 9. [CrossRef]
21. Zhou, D.; Yang, Y.; Zhu, J.; Wang, K. Intelligent Reading Recognition Method of a Pointer Meter Based on Deep Learning in a Real Environment. *Meas. Sci. Technol.* **2022**, *33*, 055021. [CrossRef]
22. Yang, Z.; Xu, Z.; Wang, Y. Bidirection-Fusion-YOLOv3: An Improved Method for Insulator Defect Detection Using UAV Image. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–8. [CrossRef]
23. Zhou, X.; Jiang, L.; Hu, C.; Lei, S.; Zhang, T.; Mou, X. YOLO-SASE: An Improved YOLO Algorithm for the Small Targets Detection in Complex Backgrounds. *Sensors* **2022**, *22*, 4600. [CrossRef] [PubMed]
24. Chen, J.; Deng, S.; Wang, P.; Huang, X.; Liu, Y. Lightweight Helmet Detection Algorithm Using an Improved YOLOv4. *Sensors* **2023**, *23*, 1256. [CrossRef] [PubMed]
25. Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Wey, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
26. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
27. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 1314–1324.
28. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
29. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11218, pp. 122–138, ISBN 978-3-030-01263-2.
30. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
31. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features From Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586.

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