

Article

FM-STDNet: High-Speed Detector for Fast-Moving Small Targets Based on Deep First-Order Network Architecture

Xinyu Hu, Defeng Kong *, Xiyang Liu, Junwei Zhang and Daode Zhang

School of Mechanical Engineering, Hubei University of Technology, Wuhan 430000, China; 19991012@hbut.edu.cn (X.H.)

* Correspondence: kongdefeng0916@gmail.com

Abstract: Identifying objects of interest from digital vision signals is a core task of intelligent systems. However, fast and accurate identification of small moving targets in real-time has become a bottleneck in the field of target detection. In this paper, the problem of real-time detection of the fast-moving printed circuit board (PCB) tiny targets is investigated. This task is very challenging because PCB defects are usually small compared to the whole PCB board, and due to the pursuit of production efficiency, the actual production PCB moving speed is usually very fast, which puts higher requirements on the real-time of intelligent systems. To this end, a new model of FM-STDNet (Fast Moving Small Target Detection Network) is proposed based on the well-known deep learning detector YOLO (You Only Look Once) series model. First, based on the SPPNet (Spatial Pyramid Pooling Networks) network, a new SPPFCSP (Spatial Pyramid Pooling Fast Cross Stage Partial Network) spatial pyramid pooling module is designed to adapt to the extraction of different scale size features of different size input images, which helps retain the high semantic information of smaller features; then, the anchor-free mode is introduced to directly classify the regression prediction information and do the structural reparameterization construction to design a new high-speed prediction head RepHead to further improve the operation speed of the detector. The experimental results show that the proposed detector achieves 99.87% detection accuracy at the fastest speed compared to state-of-the-art depth detectors such as YOLOv3, Faster R-CNN, and TDD-Net in the fast-moving PCB surface defect detection task. The new model of FM-STDNet provides an effective reference for the fast-moving small target detection task.

Keywords: mobile vision; deep learning; small target detection; SPPNet; machine learning; anchor free



Citation: Hu, X.; Kong, D.; Liu, X.; Zhang, J.; Zhang, D. FM-STDNet: High-Speed Detector for Fast-Moving Small Targets Based on Deep First-Order Network Architecture. *Electronics* **2023**, *12*, 1829. <https://doi.org/10.3390/electronics12081829>

Academic Editors: Antonio Fernández-Caballero and Byung-Gyu Kim

Received: 20 March 2023
Revised: 10 April 2023
Accepted: 11 April 2023
Published: 12 April 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

Moving small target detection has a wide range of applications in many fields. For example, in the field of autonomous driving [1], high-resolution scene photos collected by cars of pedestrian targets or traffic signs are often too small, but the accurate detection of these small moving targets is an important prerequisite for safe autonomous driving; automatic industrial inspection to locate small defects on material surfaces also illustrates the importance of small target detection [2]; UAVs move rapidly in the dangerous high-altitude environment to identify tiny fault points or foreign objects, effectively safeguarding the lives and property of operational workers [3]; modern military weaponry strikes fast-moving enemy targets with real-time precision [4]. In the real world, more sophisticated systems are deployed and fast-moving small-target detection has great value.

Mobile vision is a very important task in computer vision, and the target detection model built on deep learning theory has the feature of high detection accuracy. As the detection accuracy increases, the model is designed to be more complex, and the number of parameters increases dramatically, which greatly increases the operational burden of the supporting equipment. In the face of increasingly bloated target detection models, there

are three general solutions. The first approach is to lighten the model [5–7]. Lightweighting methods mostly sacrifice detection accuracy in exchange for a reduced amount of model parameters, thus obtaining a faster computation of the model. The second approach is to use cloud-side collaboration [8–10], which refers to placing simple processes that require real-time computation and analysis closer to the end device to ensure real-time data processing and also to reduce the risk of data transmission. This cloud-side collaboration model has high network requirements and requires sufficient memory and high-performance processors to process large amounts of data. The third approach is to build an optimized system solution locally, such as the DeepCache framework [11–14]. Deep learning engines can cache results when executing CNNs on mobile video, using the input frame contents as cache keys and inference results as cache values. This approach reduces the computation while increasing the data movement overhead and increasing the cache burden. This research focuses on real-time inspection tasks with printed circuit boards, PCB surface defects detection has three difficulties: (1) PCB surface defects are extremely small compared to the proportion of pixels in the whole image; (2) the defective part is very similar to the background part, almost indistinguishable to the naked eye; (3) PCB surface defects are usually hidden between the tiny wiring, greatly increasing the difficulty of detection.

To address the limitations of the above methods in the moving target detection problem and to consider the difficulties of PCB surface defect detection tasks, two novel networks named SPPFCSP and RepHead were proposed based on the YOLO family of algorithms [15–17] to construct a new detector model FM-STDNet. This model improves the model without reducing the detection accuracy and the computational speed, which can be competent for fast-moving small targets in real-time accurate and fast detection scenarios. More specifically, the former extracts feature of different scale sizes by connecting three Maxpool blocks of different scale sizes in series to effectively capture small targets; the latter absorbs the excellent performance of the prediction part of the YOLOX model and reconfigures the YOLO head to improve the model's operation speed, thus adapting to the detection scenario of fast-moving targets. Extensive experimental results in various environments show that the proposed FM-STDNet network exhibits superior performance in terms of both accuracies (as measured by precision and recall) and runtime speed (as measured by FPS) compared to state-of-the-art detectors and is equally robust in irregularly placed fast-running PCBs.

The rest of the paper is organized as follows. Section 2 describes the methodology in detail. Section 3 verifiably evaluates the performance of the proposed new module on the COCO2017 public dataset. Section 4 verifiably evaluates the performance of the proposed FM-STDNet detector on an independently designed PCB surface defect detection experimental platform. Section 5 concludes the paper and looks forward to future work.

2. Materials and Methods

In this section, a new network SPPFCSP based on a spatial pyramid structure is first proposed and theoretically demonstrated; then, a novel high-speed detection head, RepHead, is proposed. It should be noted that the SPPFCSP spatial pyramid pooling with the RepHead detection head can be applied to all multi-scale feature map-based target detection tasks.

2.1. A Faster Spatial Pyramid Pooling Structure for the New Network SPPFCSP

Spatial pyramidal pooling can transform the convolutional features of any scale image into the same dimension [18], which is of great importance not only for allowing CNNs to process images of arbitrary scales but also for avoiding the loss of some information due to cropping and warping operations. It also solves the problem of repeated extraction of image-related features by convolutional neural networks, greatly improving the speed of generating candidate frames and saving computational costs. In YOLOv7, the authors absorbed the CSPnet structure [19] to reduce the model computation and improve the running speed without degrading the performance of the model accuracy and designed the

SPPCSPC module to improve the network running speed. To further reduce the number of parameters of the model, as well as to meet the requirements of fast-moving object detection in practical deployment, and to learn from the SPPF structure proposed by the authors in YOLOv5 [20], a new model of spatial pyramidal pooling, SPPFCSP, with different sizes of Maxpool layers in series is proposed. The structure of the model is shown in Figure 1.

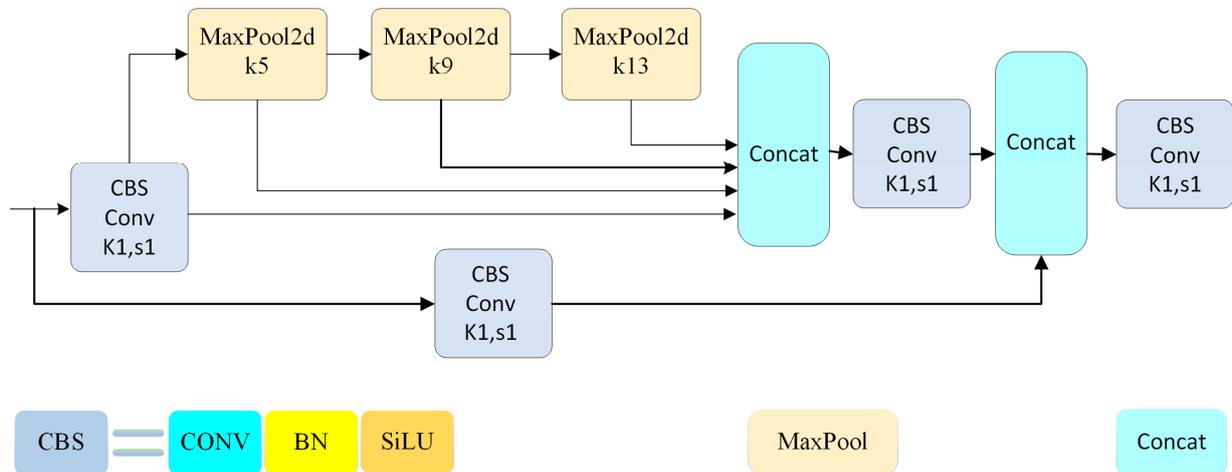


Figure 1. SPPFCSP network structure.

In Figure 1, the first branch starts with a CBS module for feature extraction, and then three Maxpool modules are connected in series to extract features of different scale sizes, 5×5 , 9×9 , and 13×13 , respectively. The feature maps extracted from the three different scale boxes are concatenated with the feature maps extracted from the 1×1 scale box, and finally, a CBS module is connected to the output. Another branch is constructed based on the CSPNet network structure, and the CSP module divides the feature layer into two parts, one is the pooling branch mentioned above, and the other part is merged with the maximum pooling feature layer after feature extraction by connecting the CBS module. This can reduce the computation by half, speed up the model operation, and improve the detection accuracy.

The following content demonstrates the feasibility of SPPFCSP. Let the input feature map size be $W_{in} \times H_{in} \times C_{in}$ and the desired pooling layer output size be n . The pooling process does not change the channel, so $C_{ot} = C_{in}$, i.e., the output size is $n \times n \times C_{out}$. The pooling layer filter size (F_w, F_h) is as in Equations (1) and (2).

$$F_w = \left\lceil \frac{W_{in}}{n} \right\rceil \tag{1}$$

$$F_h = \left\lceil \frac{H_{in}}{n} \right\rceil \tag{2}$$

The expression of stride (S_w, S_h) for the pooling layer is given in Equations (3) and (4):

$$S_w = F_w \tag{3}$$

$$S_h = F_h \tag{4}$$

According to the Equations (5) and (6) of the convolutional layer pooling layer output size:

$$W_{out} = \frac{W_{in} - F_w}{S_w} + 1 \tag{5}$$

$$H_{out} = \frac{H_{in} - F_h}{S_h} + 1 \quad (6)$$

When given any size of input and n , the output size of the pooling layer is expected to be $n \times n \times C_{out}$; i.e., proving whether Equations (7) and (8) holds:

$$W_{out} = n \quad (7)$$

$$H_{out} = n \quad (8)$$

Take $W_{out} = n$ as an example to illustrate the proof process, and $H_{out} = n$ to prove the same thing. Substitute (1) and (3) into (5) to obtain (9). Whether or not to take the upper integer needs to be discussed; here, we only look at a special case, that is, when it is exactly divisible by an integer:

$$\begin{aligned} W_{out} &= \frac{W_{in} - \lceil \frac{W_{in}}{n} \rceil}{\lfloor \frac{W_{in}}{n} \rfloor} + 1 \\ &= \frac{W_{in} - \frac{W_{in}}{n}}{\frac{W_{in}}{n}} + 1 \\ &= n \end{aligned} \quad (9)$$

As a result, the channel is consistent for different input sizes, and the n -values are consistent, so the output size is consistent; i.e., Equation (7) holds. Thus, it can be adapted to different sizes of image inputs. Assuming that each feature map gets f features and feature $f = n \times n$ size, the output of the fully connected layer is $C_{out} \times f$.

Since $W_{out} = n$ and $H_{out} = n$ are known to hold, and $f = W_{out} \times H_{out}$, therefore $f = \sum_{i=1}^n n_i^2$ holds. That is, every feature map of different sizes can be obtained by setting multiple pooling layers and automatically adjusting the filter size and step size of the pooling layer by n to obtain the same feature map, which will be concatenated to arrive at the final desired feature map size.

2.2. A High-Speed Detection Head, RepHead, Based on Structural Re-Parameterization Construction

By analyzing the Anchor base pattern [21], four main limitations are summarized: an unbalanced number of positive and negative samples, a large computational effort, poor generalization ability, and high subjectivity. Based on the above Anchor base model defects, this paper selects the anchor-free model [22] as the basic framework of the detection head, while absorbing the superior performance of the decoupled head model [23] to decouple the feature map prediction information. In YOLOX, to balance speed and performance, the authors finally used one 1×1 convolution for dimensionality reduction and two 3×3 convolutions in each of the classification and regression branches, and finally adjusted to increase some parameters in exchange for improved accuracy performance. The decrease in the speed of YOLOX in the s,m,l,x model is also due to this [24]. However, this method of compressing the number of parameters will lose some feature information, which will cause a loss of accuracy in the later classification and regression process. Based on these considerations, a non-destructive compression detection head module method is proposed to substantially improve the inference speed of detection while ensuring detection accuracy. The structure of the module is shown in Figure 2.

The 1×1 convolution is transformed into a Rep module by integrating it with a reparameterization method [25]. The concept of structural reparameterization proposed here refers to the fact that the structure at training corresponds to a set of parameters and the desired structure at inference corresponds to another set of parameters; the structure of the former can be equivalently transformed to the latter as long as the parameters of the former can be equivalently transformed to the latter. The Rep module has a multi-branch structure during the training phase, which facilitates the learning of the network and

speeds up the convergence of the network, and only one convolutional and activation function during the inference phase, which improves the inference efficiency. To match the Anchor free model, the features are decoupled into three branches after the Rep module, which contains information such as type category, the judgment of whether to include the target, and the detection frame. The prediction information of the three branches is fused once, and then all the different scales of feature prediction information are reshaped, and then a large fusion is performed to get the prediction information of feature points $N \times (cls_output + obj_output + reg_output)$.

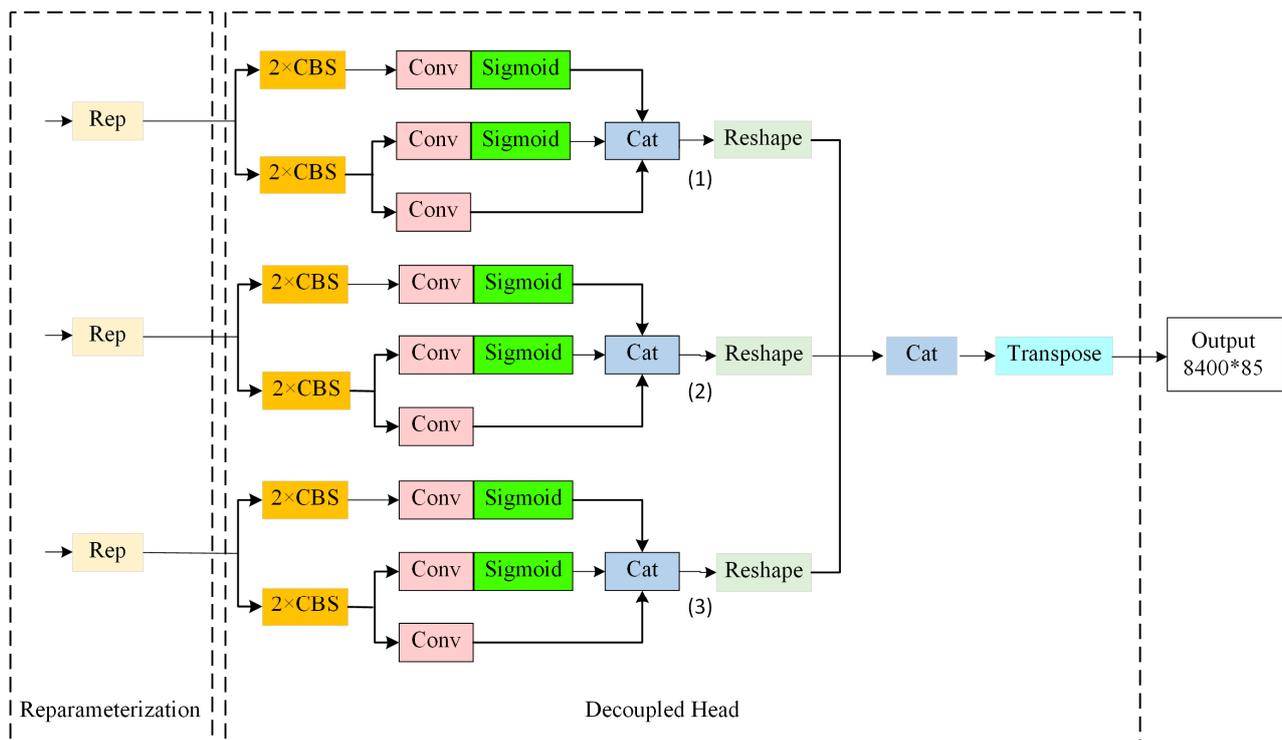


Figure 2. RepHead network structure.

In Figure 2, the size of the anchor frame corresponding to 400 frames in branch (1) is 32×32 ; the size of the anchor frame corresponding to 1600 prediction frames in branch (2) is 16×16 ; the size of the anchor frame corresponding to 6400 prediction frames in the branch (3) is 8×8 . When there is information on 8400 prediction frames, each picture also has information on the labeled target frame. At this point, the anchor frames are equivalent to bridges, the 8400 anchor frames are correlated with all the target frames on the pictures, and the positive sample anchor frames are selected. And accordingly, the positions corresponding to the positive sample anchor frames can be picked out from the positive sample prediction frames. The association method used is label assignment. The label assignment goes through two processes: preliminary screening and SimOTA [26]. The FocalLoss function [27] is selected to calculate the error between the target frame and the positive sample prediction frame, which is used to update the network parameters.

3. Evaluating the Performance of SPPFCSP and RepHead on a Benchmark Dataset

In this section, adequate ablation experiments are conducted on the most widely used MS COCO2017 target detection public dataset [28] to test the actual performance of the proposed SPPFCSP and RepHead. Firstly, the adopted dataset and experimental setup are briefly introduced, then the meaning and role of the main evaluation metrics are presented, and finally, the results of the ablation experiments are analyzed to verify the effectiveness and generalization ability of the proposed SPPFCSP and RepHead. All models experiment on the same experimental platform.

3.1. Data Set Introduction and Experimental Setup

In this work, the well-known benchmark dataset for target detection, MS COCO2017, is used for performance evaluation. MS COCO2017 is the most challenging target detection task available and has become a practical benchmark in the field of target detection. MS COCO2017 contains 164k images and 897K targets labeled from 80 categories. Moreover, it contains more small targets (whose area is less than 1% of the image) and more densely localized targets. Ablation experiments were done on the COCO2017 dataset with 118,287 and 5000 training and test samples, respectively. Details of the experimental setup of the dataset are shown in Table 1.

Table 1. Experimental setup.

Set Item	Parameter
Iteration	200
Batch size	16
Initial learning rate	0.1
Min learning rate	0.0001
Optimizer	SGD
momentum	0.937
Weight decay	5×10^{-4}
Learning rate decay type	COS
thread	4

3.2. Evaluation Indicators and Roles

The ultimate goal of the target detection task is to improve the speed of detection while ensuring accuracy. Many practical applications of target detection technology have high requirements in terms of accuracy and speed. If we disregard speed performance metrics and focus only on accuracy performance breakthroughs, the cost is higher computational complexity and more memory requirements, limiting scalability for full industry deployment. Therefore, this paper takes into account the average detection accuracy, model complexity, and model running speed in evaluating model performance. This part of the experiment uses three metrics commonly used in target detection algorithms, mAP, FLOPs, and FPS, as evaluation metrics.

3.3. Ablation Experimental Results and Discussion

Two improvement networks, SPPFCSP and RepHead, are mentioned in the method section, and the proposed two improvement networks are studied for ablation. Table 2 records the results of a series of ablation studies done on the benchmark public dataset COCO2017. Since the SPPFCSP network is based on the SPPNet improvement of YOLOv5 and the RepHead is based on the Head part improvement of YOLOX, YOLOv5-s, and YOLOX-s are selected as the baseline methods for this experiment. In Section 3.2, mAP, GFLOPs, and FPS are mentioned as evaluation metrics.

Table 2. Ablation studies on the COCO2017 dataset.

Algorithm	mAP (%)	GFLOPs	Speed (bs = 1,fps)
YOLOv5-s	36.7	17.1	85
+SPPFCSP	39.8 (+3.1↑)	15.2	91 (+6↑)
YOLOX-s	39.6 (+1.9↑)	26.8	81
+RepHead	39.7 (+0.1↑)	14.3 (−12.5↓)	108 (+27↑)

Based on the experimental results, the analysis summarizes three conclusions:

- (1) When the traditional SPP module is replaced with the SPPFCSP network, there is a 3.1% improvement in the average detection accuracy and 6 FPS in the computational speed, which indicates that SPPFCSP makes better feature fusion between local

features and global features, and tandem different scale Maxpool layers give full play to the advantages of the SPPFCSP network structure for fusing different scale features while solving the CNN network's problem of feature repetition extraction, reducing the model computation, improving the model operation speed, and increasing the detection accuracy.

- (2) The prediction part of the YOLOv5-s model is replaced by the prediction part structure of YOLOX-s, which contains four technical points Decoupled Head, Anchor Free, Label Assignment, and FocalLoss. The experimental results show that the YOLOX-s detection accuracy is 1.9% higher than YOLOv5-s, indicating that the Decoupled Head structure of the Anchor Free mode, together with the SimOTA label assignment strategy and the FocalLoss loss function, is more consistent with the feature expression of the YOLO detection framework, but the FLOPs increase, indicating that the YOLOX_Head model increases the model complexity and is not conducive to real-time detection of fast-moving objects.
- (3) Structural reparameterization of the YOLOX head part is constructed to obtain the Rep_Head module, and the experimental results show that mAP is improved by 0.1%, FLOPs are decreased by 12.5, and the FPS index is significantly improved by 27, indicating that structural reparameterization can maintain the detection accuracy of the original network while reducing the model complexity and improving the network operation speed. Compared with other methods that simply reduce the number of model parameters to reduce model complexity, the structural reparameterization in the prediction part of the construction does not compress the number of real model parameters, thus allowing the model to maintain the detection accuracy while significantly increasing the model computation speed.

4. Verification of FM-STDNet Network Detection on Fast-Moving Tiny Targets

In this section, the proposed FM-STDNet network is used to detect surface defects on a moving PCB board to verify the detection performance of the framework for fast-moving tiny targets.

4.1. FM-STDNet Network Framework Structure

The FM-STDNet model framework is shown in Figure 3. The FM-STDNet model uses the classical YOLO series organization architecture, which consists of input, feature extraction, neck, and prediction parts.

Using the YOLOX network as the base architecture, the last SPP layer of the backbone feature extraction network is replaced by the SPPFCSP module proposed in Section 2.2. This module enables the network to input images of arbitrary size while keeping the output feature map size unchanged on the one hand, reduces the model computation, and improves the model running speed on the other hand. In the prediction part, we take advantage of the Decoupled Head structure with better expressiveness, Anchor Free mode with fewer parameters, SimOTA with more reasonable label assignment, and FocalLoss loss function with faster convergence in YOLOX, and combine the idea of structure reparameterization construction for lossless compression to design the RepHead prediction module to replace the original YOLOX detection head.

4.2. FM-STDNet Evaluation Results and Discussion

To evaluate the performance of the proposed FM-STDNet framework, the network model was trained on the PCB surface defect public dataset. The PCB surface defect dataset [29] published by the Open Laboratory of Human-Computer Interaction, Peking University, has 693 images with 3–5 defects on each image. The defects include six types: missing hole, mouse bite, open circuit, short, spur, and spurious copper. The defect images in the original dataset are high resolution and, for such a small dataset, data enhancement techniques are used before data training. The images are then cropped into 600×600 sub-images, resulting in a training and test set of 9920 and 2508 images,

respectively. Most of the defect targets are tiny targets and the defects blend with the background to form a low contrast, which is challenging for accurate defect localization and classification.

4.2.1. Experimental Platform

All training and evaluation were performed on the open source toolbox PyTorch via the Psycharm platform on a computer with a 12th Gen Intel® Core™ i5-12600KF 3.70 GHz CPU and 16 GB of installed memory, and an NVIDIA GeForce RTX 3070 GPU on Windows 10 (with 8 GB of memory) to run FM-STDNet and the other target detection models involved in the comparison experiments. The experimental parameters of the models were set as in Table 1 in Section 3.1.

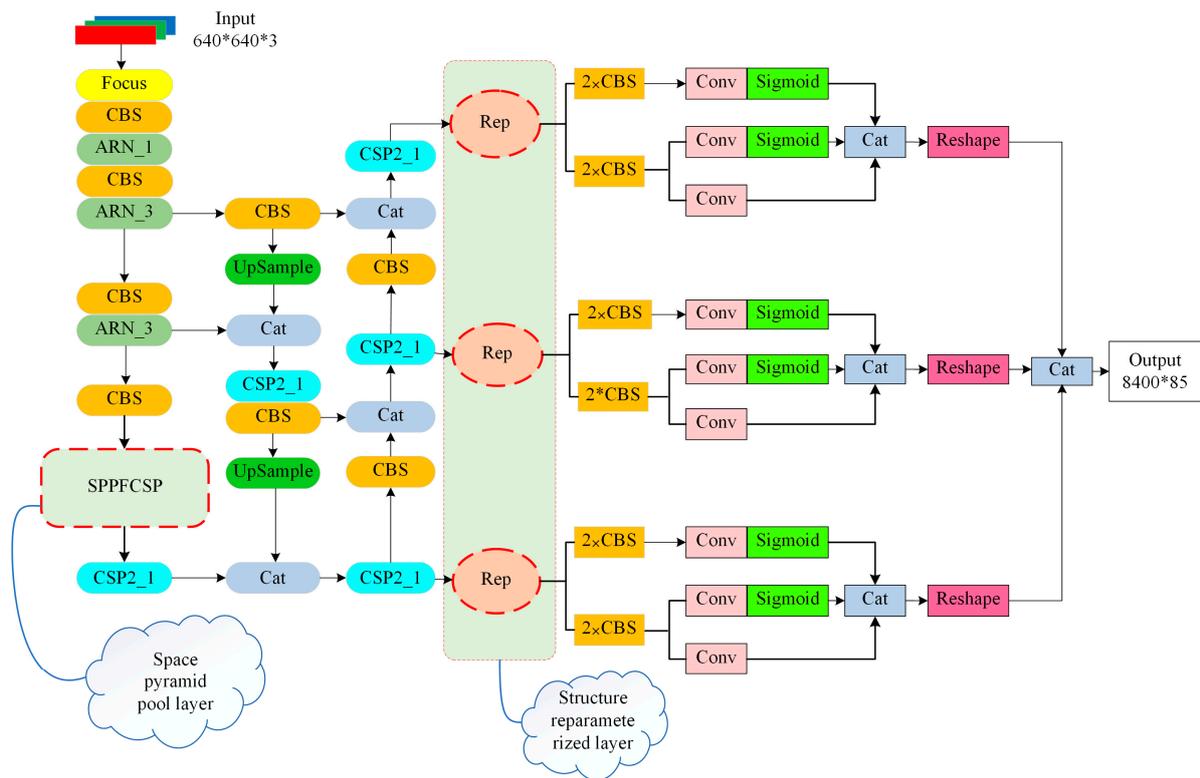


Figure 3. FM-STDNet network structure.

4.2.2. Comparison of Detection Results between FM-STDNet and Other Five Excellent Models

To further verify the detection performance of the proposed FM-STDNet algorithm on high pixel low contrast tiny size targets, comparison experiments were done on the public dataset PCB surface defect images at evaluation metrics including mAP@0.50 and FPS. The test results are shown in Table 3.

Table 3. Detection results of FM-STDNet and other 5 excellent models.

Algorithm	mAP@0.50 (%)	Speed (bs = 1,fps)
SSD	95.70	46
Faster R-CNN	91.74	37
YOLOv3	97.50	45
TDD-Net	98.24	62
YOLOX-s	98.96	90
FM-STDNet	99.85	116

As can be seen from Table 3, by observing the results of the other five excellent algorithms tested on the public dataset of PCB surface defects, it was found that the FM-STDNet model obtained the highest detection accuracy of 99.85% and the fastest computational speed of 116 FPS. In terms of the mAP@0.50 metric, FM-STDNet was 0.89% more accurate than the best-performing YOLOX-s model for detection and 8.11% more accurate than the worst-performing Faster R-CNN, which is a very clear advantage. In terms of FPS metrics, FM-STDNet ran at the highest 116 FPS, which was much faster than the other five detectors. This indicates that the SPPFCSP module reduces the computation of the model and improves the model's running speed; the RepHead module optimizes the feature expression of the model and improves the detection accuracy; and the unique re-parameterization construction makes the model obtain a higher computation speed during the inference process.

4.2.3. Performance Evaluation of FM-STDNet

To verify that the proposed FM-STDNet model has the same superior performance on actual fast-moving small targets, a PCB surface defect real-time detection experimental platform was designed with the structure shown in Figure 4.



Figure 4. PCB surface defects real-time detection experimental platform.

The main structure of the experiment platform consists of three parts: an object transmission unit, a detection unit, and a data processing and display unit. The object transfer unit has an adjustable transfer speed to test the performance of the algorithm in detecting targets at different moving speeds; the detection unit contains a CCD industrial camera and a horizontal light source; the data processing and display unit is a desktop computer with Windows 10 and an NVIDIA GeForce RTX 3070 GPU graphics card.

A PCB with artificial surface defects containing all defects (two defects of each type for a total of twelve defects) is produced, and according to the definition of the international organization SPIE [30], a small target is an area of fewer than 80 pixels in a 256×256 image, i.e., less than 0.12% of 256×256 . As shown in Figure 5, there are small, medium, and large defect detail maps in the high-pixel image (3056×2464).

This PCB test prototype is placed on the designed PCB surface defect real-time inspection experimental platform conveyor belt, and two levels of conveyor belt running speeds are set: low speed (2 cm/s) and high speed (20 cm/s). By changing the speed single factor, the detection effect of the YOLOX-s and FM-STDNet model is tested, and the detection results are shown in Figure 6. The horizontal axis in Figure 6 shows the detection results for different running speeds of the PCB (Low speed and High speed), and the vertical axis shows the detection results for different models (YOLOX and FMSTDNet).

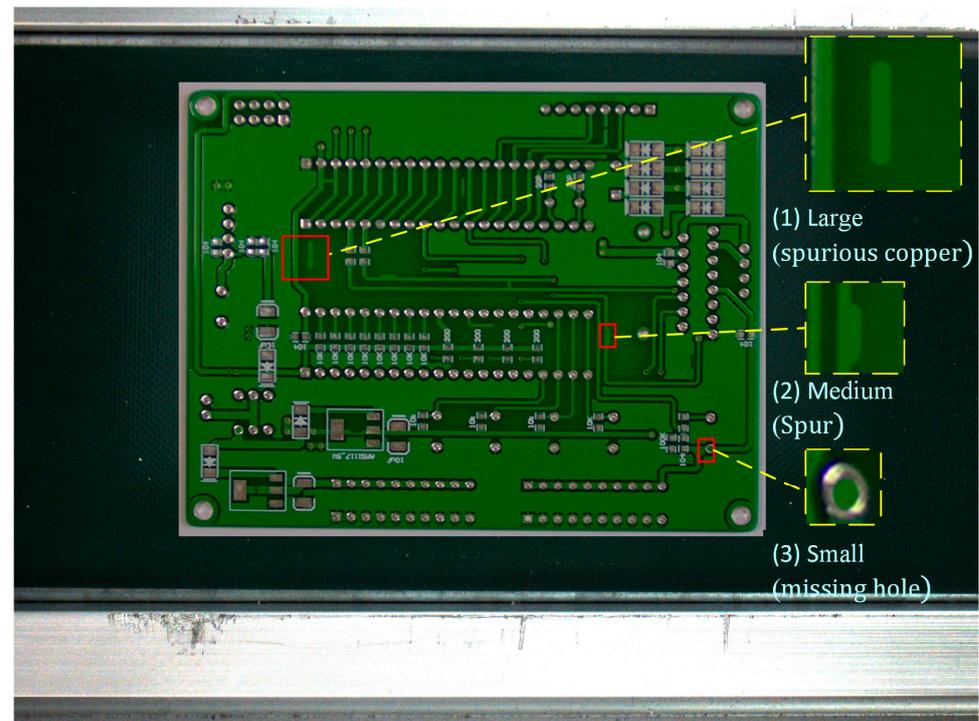


Figure 5. Details of the small, medium, and most defects.

As observed in Figure 6, the actual inference speeds of YOLOX (one of the best single-stage detectors available) and FM-STDNet models in the low-speed (2 m/s) target operation phase were 91.13 FPS and 116.07 FPS, respectively, indicating that FM-STDNet had a faster data processing speed than YOLOX. The total number of targets detected by both models was 12, indicating that during the target low-speed moving phase, the target moving speed met the model calculation speed and both YOLOX and FM-STDNet models could successfully detect all targets. In the fast (20 cm/s) target movement phase, the calculation speeds of the YOLOX and FM-STDNet models changed slightly, but at this time, YOLOX detected 11 targets, while the FM-STDNet model detected all 12 targets correctly, and one target was missed by YOLOX model was a small target. This indicates that the FM-STDNet model had better detection performance for fast-moving small targets.

On the same experimental platform, the PCB test prototype was allowed to run at 20 cm/s to test several of the best target detection models, including YOLOv3, Faster R-CNN, and TDD-Net. The experimental test results are shown in Figure 7.

Figure 7 shows the detection results of the proposed FM-STDNet model compared to the other three best target detection models. The observed results show that the actual running speeds of the three models, Faster R-CNN, YOLOv3, and TDD-Net, are 37.28 FPS, 45.78 FPS, and 90.78 FPS, respectively, and the total number of detected targets is 7, 8, and 11, respectively. All three models showed missed detections, but the actual running speed of the FM-STDNet model was 115.19 FPS and the total number of detected targets was 12, indicating that the FM-STDNet model had the highest actual inference speed and accuracy on the fast-running small target detection task.

In actual PCB production, a quality sampling inspection with a failure can seriously affect the order delivery and cause huge economic loss to the PCB producer. Therefore, the robustness of the model is particularly important, and the size of the area covered by the precision-recall curve needs to be calculated. Figure 8 shows the precision-recall (PR) curves for six defects on the PCB defect image at IOU = 0.5. Among them, the PR curve represents the relationship between precision and recall and is usually set with recall as the horizontal coordinate and precision as the vertical coordinate. The larger the value of both recall and precision, the better the performance of the model, i.e., the closer the two curves are to the upper right corner of the image the better the performance.

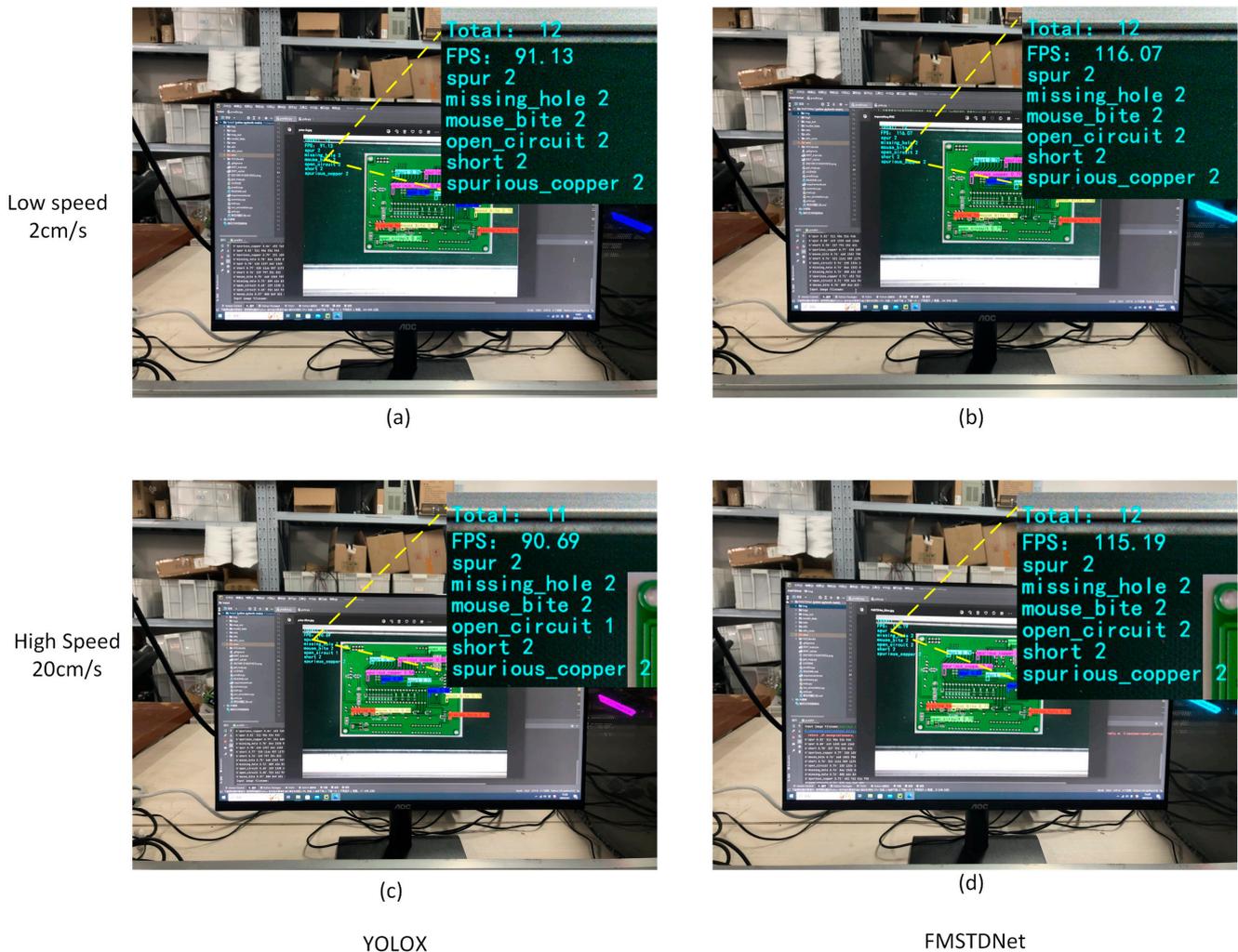
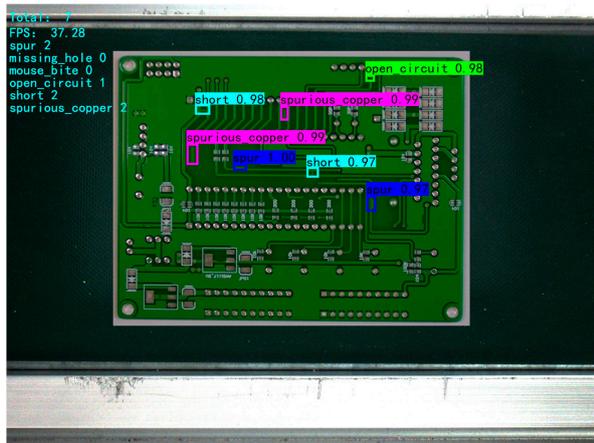
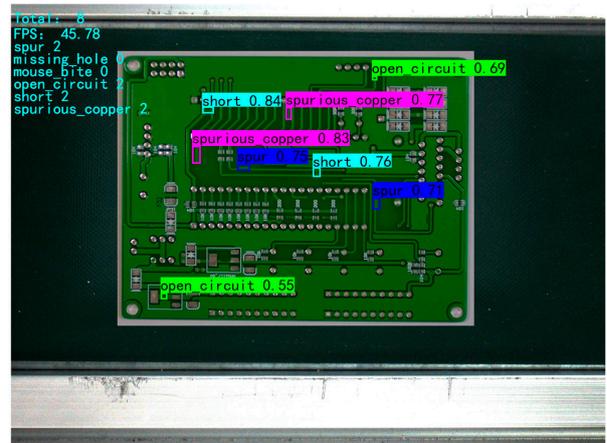


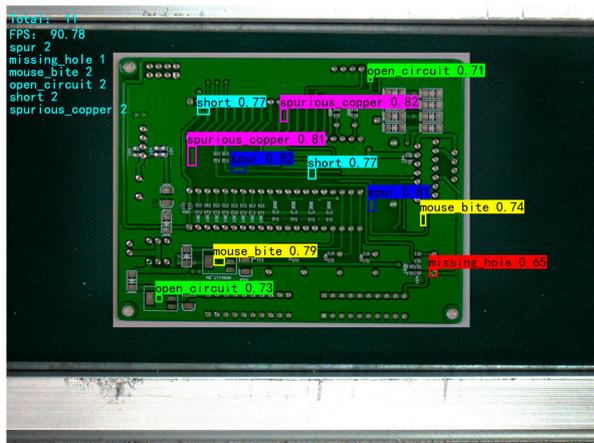
Figure 6. Comparison of YOLOX and FM-STDNet detection results. (a) Detection results of YOLOX model at 2 m/s moving speed; (b) Detection results of FMSTDNet model at 20 m/s moving speed; (c) Detection results of YOLOX model at 2 m/s moving speed; (d) Detection results of FMSTDNet model at 20 m/s moving speed.



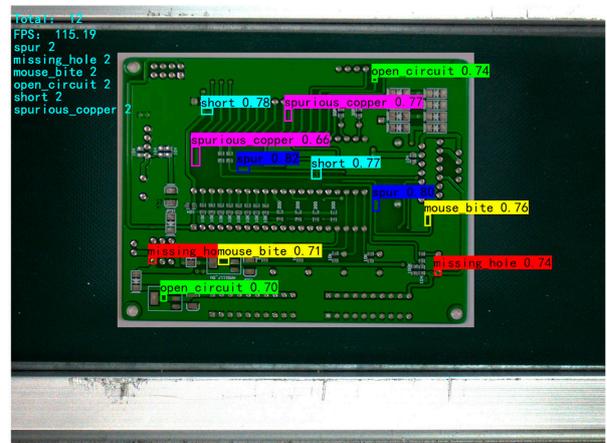
(1) Faster R-CNN



(2) YOLOv3

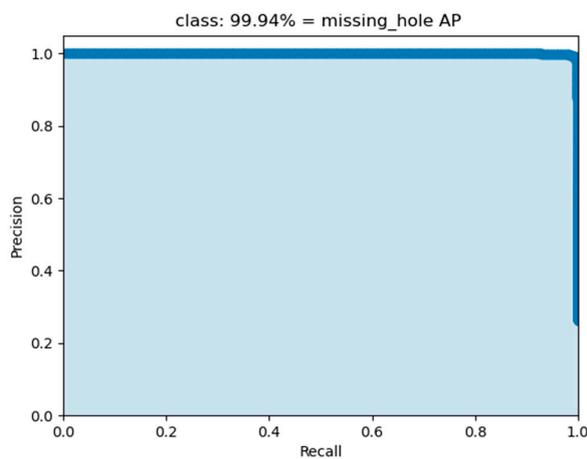


(3) TDD-Net

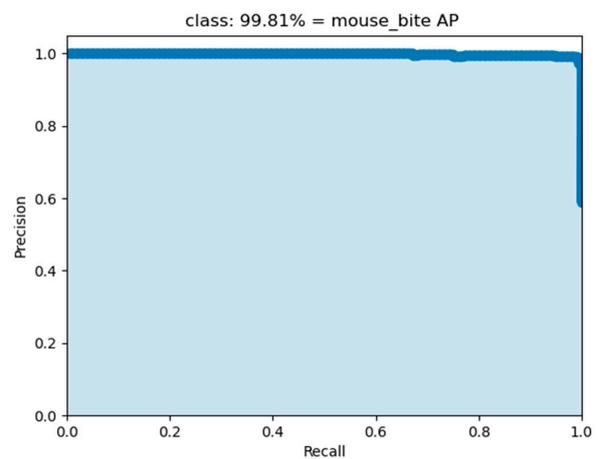


(4) FMSTDNet

Figure 7. Comparison of detection performance of FM-STDNet model with Faster R-CNN, YOLOv3, and TDD-Net models on 20 m/s mobile PCB.



(a) minssing_hole AP =99.94%



(b) mouse_bite AP =99.81%

Figure 8. Cont.

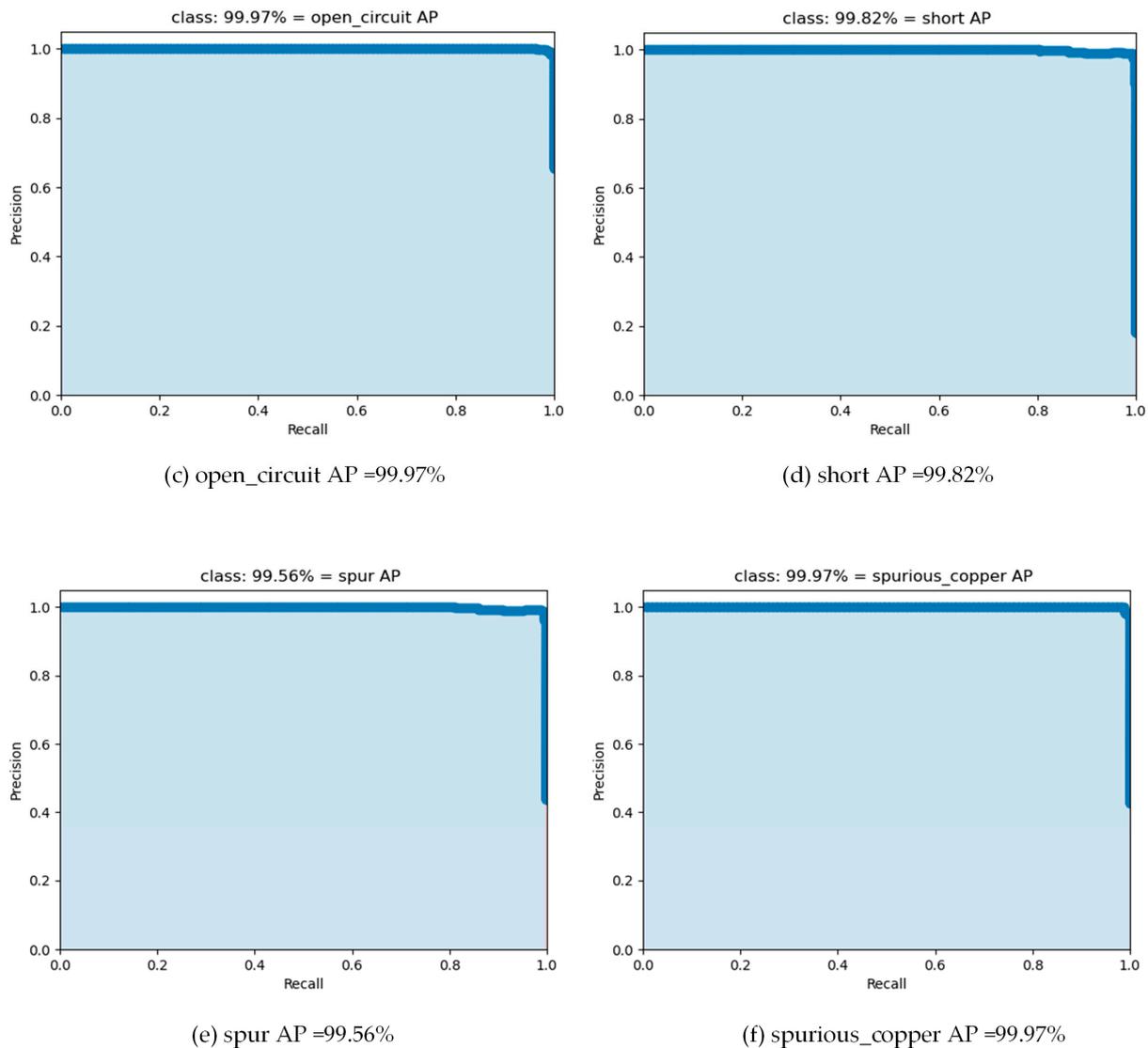


Figure 8. PCB defect precise recall (PR) curve at AP@0.5.

The exact recall curves for each of the six defective targets are calculated in Figure 8, and it can be seen that the area under the PR curves of the six targets is almost close to 1, indicating that the FM-STDNet model has high detection accuracy and recall for all targets and shows good robustness.

5. Conclusions and Limitation Statement

In this paper, the defect detection performance of a PCB surface defect detection system was investigated. This task is challenging because the PCB defects are very small compared to the whole PCB area, and in addition, they move very fast and must be detected in real-time. Based on the SPPCSPC module and YOLOX model, two new modules, SPPFCSP and RepHead, are proposed. The FM-STDNet fast-moving small target detection model is constructed with YOLOX as the base structure. On the self-designed experimental platform of the PCB surface defect detection system, FM-STDNet achieves an average detection accuracy of 99.85% in a PCB running at high speed and completes the real-time detection task with a 116FPS actual detection refresh rate. The performance of the proposed FM-STDNet model is compared by testing with the current state-of-the-art detectors, including YOLOv3, Faster R-CNN, and TDD-Net. The results show that the FM-STDNet detector can

achieve high detection accuracy with very efficient computational performance compared to these methods, which proves the superiority and effectiveness of the proposed model.

The current study also has the limitation that FM-STDNet achieves high efficiency and high accuracy detection of fast-moving small targets, relying heavily on superior hardware performance, such as high-speed industrial camera CCDs and high-performance GPUs. However, many practical applications are embedded and cannot simply be configured with better data acquisition units and processing units based on consideration of the actual cost and working environment, so it is important to consider the adaptability of the detector in low-configuration equipment.

Author Contributions: X.H. and D.K. completed the work and contributed to the writing of the manuscript. X.L. and J.Z. conceived the project and provided the research methodology. D.Z. conducted the survey and data management for the project. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 61976083); Hubei Province Key R&D Program of China (No. 2022BBA0016).

Data Availability Statement: The code and datasets involved in this study are publicly accessible and downloadable on the <https://github.com/jackong180/FMSTDNet> (accessed on 9 March 2023).

Acknowledgments: The authors would like to thank Xinyu Hu for his early proposal for this manuscript.

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

References

1. Liu, W.; Quijano, K.; Crawford, M.M. YOLOv5-Tassel: Detecting Tassels in RGB UAV Imagery with Improved YOLOv5 Based on Transfer Learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 8085–8094. [[CrossRef](#)]
2. Zeng, N.; Wu, P.; Wang, Z.; Li, H.; Liu, W.; Liu, X. A Small-Sized Object Detection Oriented Multi-Scale Feature Fusion Approach with Application to Defect Detection. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3997. [[CrossRef](#)]
3. Sadykova, D.; Pernebayeva, D.; Bagheri, M.; James, A. IN-YOLO: Real-Time Detection of Outdoor High Voltage Insulators Using UAV Imaging. *IEEE Trans. Power Deliv.* **2020**, *35*, 1599–1601. [[CrossRef](#)]
4. Gao, K.; Xiao, H.; Qu, L.; Wang, S. Optimal interception strategy of air defence missile system considering multiple targets and phases. *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.* **2022**, *236*, 138–147. [[CrossRef](#)]
5. Haase, D.; Amthor, M. Rethinking Depthwise Separable Convolutions: How Intra-Kernel Correlations Lead to Improved MobileNets. *IEEE Xplore* **2020**, *426*, 14588–14597. [[CrossRef](#)]
6. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. IEEE GhostNet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Electr Network, Seattle, WA, USA, 14–19 June 2020; pp. 1577–1586.
7. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. IEEE Learning Transferable Architectures for Scalable Image Recognition. In Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8697–8710.
8. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [[CrossRef](#)]
9. Ren, J.; He, Y.; Yu, G.; Li, G.Y. IEEE Joint Communication and Computation Resource Allocation for Cloud-Edge Collaborative System. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Marrakech, Morocco, 15–19 April 2019.
10. Chadwick, D.W.; Fan, W.; Costantino, G.; de Lemos, R.; Di Cerbo, F.; Herwono, I.; Manea, M.; Mori, P.; Sajjad, A.; Wang, X.-S. A cloud-edge based data security architecture for sharing and analysing cyber threat information. *Future Gener. Comput. Syst. Int. J. Escience* **2020**, *102*, 710–722. [[CrossRef](#)]
11. Xu, M.; Zhu, M.; Liu, Y.; Lin, F.X.; Liu, X. ACM DeepCache: Principled Cache for Mobile Deep Vision. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom), New Delhi, India, 29 October–2 November 2018; pp. 129–144.
12. Pang, H.; Liu, J.; Fan, X.; Sun, L. Toward Smart and Cooperative Edge Caching for 5G Networks: A Deep Learning Based Approach. In Proceedings of the 26th IEEE/ACM International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018.
13. Narayanan, A.; Verma, S.; Ramadan, E.; Babaie, P.; Zhang, Z.-L. Making Content Caching Policies ‘Smart’ using the DEEPCACHE Framework. *ACM SIGCOMM Comput. Commun. Rev.* **2018**, *48*, 64–69. [[CrossRef](#)]
14. Zhang, X.; Zhou, Y.; Wu, D.; Hu, M.; Zheng, X.; Chen, M.; Guo, S. Optimizing Video Caching at the Edge: A Hybrid Multi-Point Process Approach. *Ieee Trans. Parallel Distrib. Syst.* **2022**, *33*, 2597–2611. [[CrossRef](#)]

15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. IEEE You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 779–788.
16. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [[CrossRef](#)]
17. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the 34th AAAI Conference on Artificial Intelligence/32nd Innovative Applications of Artificial Intelligence Conference/10th AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 12993–13000.
18. Diakogiannis, F.I.; Waldner, F.; Caccetta, P.; Wu, C. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *Isprs J. Photogramm. Remote Sens.* **2020**, *162*, 94–114. [[CrossRef](#)]
19. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.H. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Electr Network, Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580.
20. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [[CrossRef](#)]
21. Wei, L.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Cheng-Yang, F.; Berg, A.C. SSD Single Shot MultiBox Detector. *Arxiv* **2016**, *2*, 21–37. [[CrossRef](#)]
22. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. IEEE CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6568–6577.
23. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q.; Soc, I.C. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the 18th IEEE/CVF International Conference on Computer Vision (ICCV), Electr Network, New York, NY, USA, 11–17 October 2021; pp. 2778–2788.
24. Ji, W.; Pan, Y.; Xu, B.; Wang, J. A Real-Time Apple Targets Detection Method for Picking Robot Based on ShufflenetV2-YOLOX. *Agriculture* **2022**, *12*, 856. [[CrossRef](#)]
25. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets Great Again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Electr Network, Piscataway, NJ, USA, 19–25 June 2021; pp. 13728–13737.
26. Daogang, P.; Ming, G.; Danhao, W.; Jie, H. Anomaly Identification of Critical Power Plant Facilities Based on YOLOX-CBAM. In Proceedings of the 2022 Power System and Green Energy Conference (PSGEC), Shanghai, China, 25–27 August 2022; pp. 649–653. [[CrossRef](#)]
27. Dong, R.; Pan, X.; Li, F. DenseU-Net-Based Semantic Segmentation of Objects in Urban Remote Sensing Images. *IEEE Access* **2019**, *7*, 65347–65356. [[CrossRef](#)]
28. Liang, S.; Wu, H.; Zhen, L.; Hua, Q.; Garg, S.; Kaddoum, G.; Hassan, M.M.; Yu, K. Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25345–25360. [[CrossRef](#)]
29. Ding, R.; Dai, L.; Li, G.; Liu, H. TDD-net: A tiny defect detection network for printed circuit boards. *Caai Trans. Intell. Technol.* **2019**, *4*, 110–116. [[CrossRef](#)]
30. Ball, J.E.; Anderson, D.T.; Chan, C.S. Comprehensive survey of deep learning in remote sensing: Theories, tools, and challenges for the community. *J. Appl. Remote Sens.* **2017**, *11*, 2609. [[CrossRef](#)]

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