

Received 19 October 2022, accepted 27 November 2022, date of publication 6 December 2022,  
date of current version 20 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3227205

## RESEARCH ARTICLE

# LFF-YOLO: A YOLO Algorithm With Lightweight Feature Fusion Network for Multi-Scale Defect Detection

XIAOHONG QIAN<sup>1</sup>, XU WANG<sup>1</sup>, SHENGYING YANG<sup>1,2</sup>, (Member, IEEE), AND JINGSHENG LEI<sup>1</sup>

<sup>1</sup>Department of Electronic Information Engineering, Zhejiang University of Science and Technology, Zhejiang 310023, China

<sup>2</sup>Zhejiang Dingli Industry Company Ltd., Lishui, Zhejiang 321400, China

Corresponding author: Shengying Yang (syyang@zust.edu.cn)


This work was supported in part by the Natural Science Foundation of Xinjiang Uygur Autonomous Region under Grant 2022D01C349, and in part by the Natural Science Foundation of China under Grant 61972357.

**ABSTRACT** The detection of defects is indispensable in industrial production. Surface defects have different scales. Both minimal flaws and significant scratches may appear on the same product. The standard method uses a multi-scale feature fusion network, introducing many parameters that may reduce the inference speed. In actual industrial production scenarios, inference speed and accuracy play an equally important role. Therefore we propose an algorithm to effectively improve the detection speed while improving the detection accuracy. The model proposed in this paper called “YOLO with lightweight feature fusion network (LFF-YOLO).” First, we use ShuffleNetv2 as a feature extraction network to reduce the number of parameters. Then, to improve the efficiency of multi-scale feature fusion, we propose the lightweight feature pyramid network (LFPN). Considering that the fixed receptive field is difficult to adapt to the defects of different scales, it may lead to the difficulty of model convergence and seriously affect the detection performance. Therefore, we propose the adaptive receptive field feature extraction (ARFFE) module, which weights the multi-receptive field channels to generate multi-receptive field information. In addition, focal loss is used to solve the problem of imbalance between positive and negative samples. Finally, we conducted experiments on NEU-DET (79.23% mAP), Peking University printed circuit board defect dataset (93.31% mAP), and GC10-DET (59.78% mAP), respectively. Extensive experiments show that our proposed method achieves optimal detection speed compared with the prevailing methods, and the detection accuracy of our method is also highly competitive. We open-source our code in the following URL: <https://github.com/syyang2022/LFF-YOLO>

**INDEX TERMS** Convolutional neural network, defect detection, feature fusion, lightweight network.

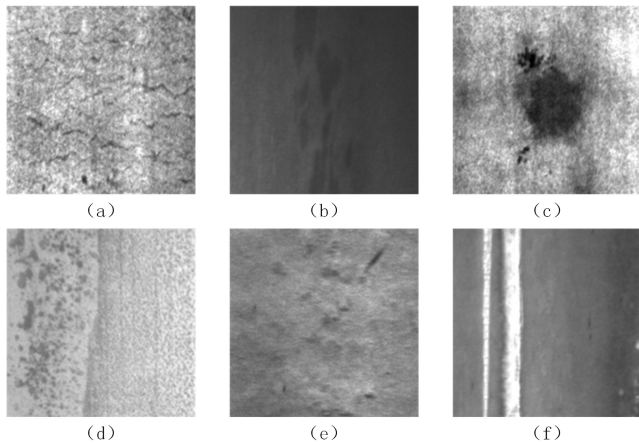
## I. INTRODUCTION

The detection of defects is indispensable in industrial production, and the detection of blemishes is a vital part of production. The use of manual methods for defect detection can lead to inefficient detection and subjective factors affecting detection accuracy. Recently, defect detection methods based on computer vision technology have gradually replaced manual defect detection. Traditional computer vision surface defect detection methods are mainly feature-based. This

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Martalo .

approach relies on manually designed algorithms to extract defect features, resulting in poor robustness and generalization of the model. Deep learning methods compensate for this shortcoming. Convolutional neural networks can capture high-level semantic features, and models have stronger robustness and generalization capabilities than traditional methods. Convolutional neural networks have become a very important method in industry [1], [2], [3].

With the rapid development of deep learning techniques, many excellent target detection algorithms have emerged, and they have been applied in the field of defect detection. Second-stage target detection algorithms such as R-CNN [4],



**FIGURE 1.** Six types of defects in NEU-DET dataset (a) Cr; (b) In; (c) Pa; (d) Ps; (e) Rs; (f) Sc.

Fast R-CNN [5], and Faster R-CNN [6], are based on area suggestion frames allowing for improved detection accuracy at the expense of detection speed. One-stage target detection algorithms such as YOLO [7], [8], [9], [10], SSD [11], RetinaNet [12], simplify the network design by using only one network to classify and localize the target, thus significantly improving the detection speed. The one-stage detection algorithm is more suitable to meet the demand for the real-time detection task. YOLOv3 [9], as a classical one-stage network, is the most widely used in industrial scenarios due to its stability. Therefore, we propose an inspection method based on YOLOv3 and improve it to make it easier to deploy on terminal devices.

Metal surface defects, such as steel surface defects (Figure 1), are more difficult to detect because of the large scale. In the feature extraction network, features extracted by the shallow layer have richer fine-grained information, such as color, texture, and other details, which can help the model identify the type of defects. Moreover, features in deeper layers mainly contain semantic information, which is the critical information for locating the defect. YOLOv3 uses the FPN [13] network to fuse multi-scale information, utilizing three feature layers for detecting small targets but not fusing the upper layer feature information for detecting large targets. PANet [14] uses a bidirectional link structure so that each detection head utilizes information from multiple feature layers, which improves the detection accuracy of large targets. However, neither FPN nor PANet is a network designed for industrial inspection, so the inference speed factor is not considered. Due to many parameters, most of the existing detection models have high deployment costs. We hope to design a model with fewer parameters and better performance based on a mature framework that has been widely used. In this paper, we propose a more efficient feature fusion network LFPN for defect detection tasks requiring real-time performance, which improves detection accuracy by introducing fewer parameters.

Lightweight networks such as ShuffleNet [15], [16], MobileNet [17], [18], and GhostNet [19] have enabled feature

extraction networks to reduce the amount of parameters without reducing accuracy. ShuffleNetv2 is an improvement of ShuffleNetv1, which considers both model parameters and the impact of memory access on inference speed. Therefore, compared with other lightweight networks, it has faster inference speed in practical applications. In this paper, ShuffleNetv2 is chosen as the feature extraction network of the model, and an adaptive receptive field feature extraction module is designed to improve the feature extraction capability.

We propose a more lightweight multi-scale feature fusion network. An adaptive receptive field feature extraction module is used to increase the feature extraction capability of the network while using the k-means algorithm to cluster the anchor frame to obtain a more reasonable anchor frame design. In addition, focal loss is used to address the problem of unbalanced positive and negative sample categories. The main contributions of this article are as follows.

- 1) We propose a new model to solve the problem in industries where metal surface defects span extremely and defect detection's low efficiency. The baseline model is the most widely used object detection model, YOLOv3.
- 2) In order to make the model easier to deploy to the detection terminal, a more lightweight network ShuffleNetv2 was used to extract defect features, but this introduced a problem that the network receptive field was fixed. Therefore, we propose the ARFFE module to increase the receptive field of the feature extraction network so that the model can extract defect information more effectively under the appropriate receptive field.
- 3) Considering the importance of detection speed in practical application scenarios, we propose a novel lightweight feature fusion network LFPN for multi-scale defects on metal surfaces, which can effectively fuse multi-scale features under the premise of introducing a few parameters, thus improving detection accuracy.
- 4) Experiments on the open defect detection dataset NEU-DET [20] show that the proposed method can detect defects quickly and effectively, which proves that the proposed method is superior to other methods in defect detection scenarios. The generalization ability of the proposed method is verified on the printed circuit board defect dataset [21] and steel plate surface defect dataset GC10-DET [22].

The rest of the article is structured as follows: related work presented in Section 2 and our proposed approach, including the lightweight feature fusion network and the adaptive perceptual field feature extraction module, presented in Section 3, experiments on NEU-DET printed circuit board open dataset and steel plate surface defect dataset GC10-DET are done in Section 4 to verify the method's validity. Finally, conclusions are given in Section 5.

## II. RELATEDWORK

The primary defect detection methods currently used in the industry include traditional methods and deep learning-based methods.

### A. TRADITIONAL METHODS

Traditional defect detection methods mainly extract image features by image preprocessing, such as histogram equalization, grayscale binarization, filtering and denoising. Subsequently, classification detection of defects was accomplished using morphology, Fourier transforms, Gabor transforms, and machine learning methods. For example, Prasitmeeboon [23] used color histogram and SVM to detect particle board defects and used thresholding and smoothing techniques to localize the faults. Chang et al. [24] implemented defect detection on the camera lens surface based on polar coordinate transform, Hough circle transform, weighted Sobel filter, and SVM. Wang and Zuo [25] used Fourier transform and Hough transforms to reconstruct the magnet surface image and obtained the defect information by comparing the grayscale difference between the reconstructed image and the original image to detect defects. These traditional methods require manual feature extraction and have poor robustness.

### B. DEEP LEARNING METHODS

In recent years, the rapid development of deep learning has led to its increasingly widespread application in defect detection. Compared with traditional methods, deep learning methods do not need to extract features manually but directly through learning data update parameters to automatically extract features and feed them into subsequent networks for classification and localization prediction. It avoids the complex process of manually designing algorithms and has a very high level of robustness and accuracy.

The currently available target detection algorithms can be divided into one-stage and two-stage networks. Two-stage networks, such as Faster-RCNN, were proposed in 2016 to improve R-CNN and Fast-RCNN. He uses the RPN network instead of the previous selective search to train the input feature map to output a series of candidate regions with initial object classification probabilities for more accurate localization of objects, resulting in improved network speed and accuracy. Zhao et al. [26] improved the traditional Faster-RCNN by reconstructing the network structure using multi-scale feature fusion and replacing part of the convolution with deformable convolution. The network was used for steel surface defect detection and reached 75.2% mAP on the NEU-DET public dataset. Cha et al. [27] applied Faster-RCNN for concrete crack and steel corrosion defect detection. Su et al. [28] designed a complementary attention network to exploit the advantages of spatial location features and channel features while suppressing background noise features and embedding them into Faster-RCNN to detect solar cell electroluminescence images. The two-stage network has satisfactory results in terms of detection

accuracy. It is challenging to meet the requirements in real-time scenarios such as industrial defect detection due to the detection efficiency problem, so the single-stage target detection network has received more attention.

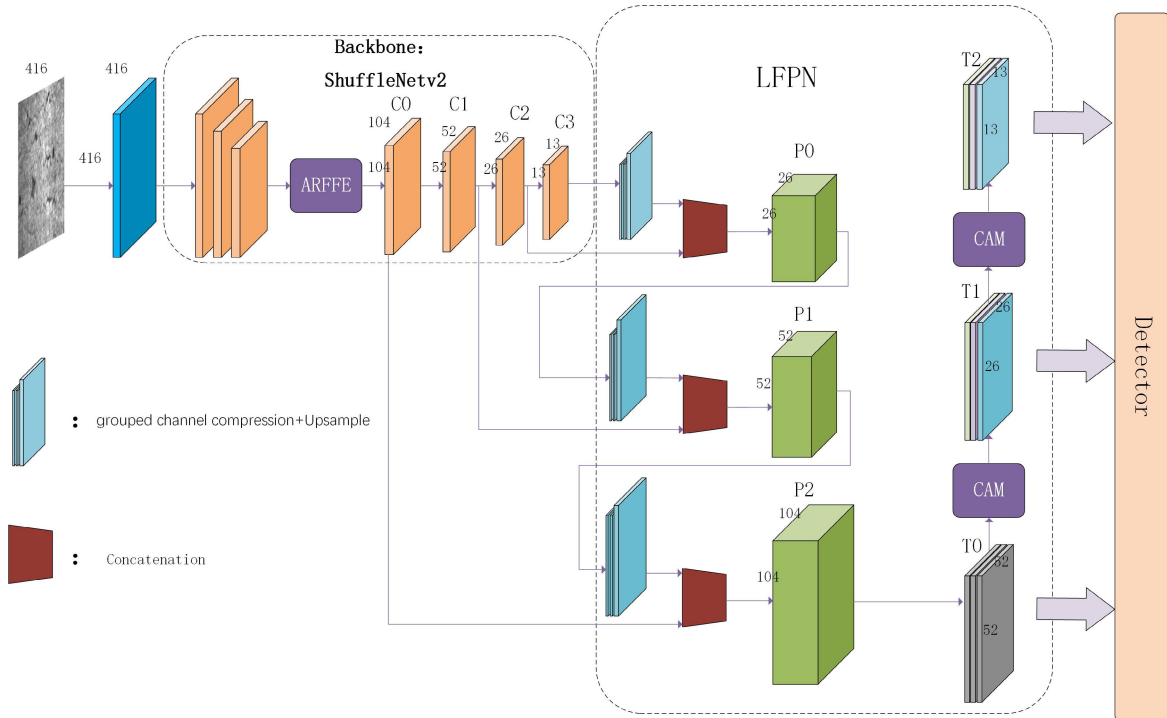
Single-stage target detection networks such as YOLOv3, SSD, and RetinaNet are more advantageous in inference speed due to their more straightforward structure than two-stage networks. With the development of single-stage networks in recent years, the gap in detection accuracy compared to two-stage networks no longer exists. Yin et al. [29] used YOLOv3 to detect sewer pipe defects and achieved 85.37% mAP. Zhang et al. [30] improved the original YOLOv3 by introducing a new migration learning method for detecting concrete bridge defects, and its performance was improved by 13% compared to the original YOLOv3. Yu et al. [31] improved YOLOv4-CSP based on the problem of small targets for industrial defect detection. They proposed an efficient stepped pyramidal network for fusing multi-scale features, thus improving the detection accuracy of small objects. Wang and Cheung [32] improved the model based on center-net by adding count loss for detecting defects generated in the Additive manufacturing process. Since the comprehensive performance of the single-stage detector is higher than that of the two-stage sensor, it is more widely used in industrial defect detection. In this paper, we choose YOLOv3 as the benchmark model and improve it to make it more suitable for industrial defect detection scenarios.

## III. METHOD

In this section, we describe the method we used in detail, and the network structure is shown in Figure 2. ShuffleNetv2 is used as the backbone feature extraction network. An adaptive receptive field feature extraction module is inserted into the backbone network to obtain different receptive fields for defects of various sizes. Then a lightweight feature pyramid network is constructed to fuse defect features of different scales more efficiently. In addition, we use the K-means algorithm to cluster the size of anchor frames and focal loss to solve the problem of positive and negative sample imbalance.

### A. LIGHTWEIGHT FEATURE PYRAMID NETWORK

DarkNet-53, as the feature extraction network of the original YOLOv3, improves the detection accuracy due to the stacking of a large number of residual blocks, but at the same time, increases the number of parameters resulting in a slower inference speed. In this paper, ShuffleNetv2 is used as the backbone feature extraction network to obtain faster inference speed and accuracy. For lightweight feature extraction networks, most network designs rely on depthwise separable convolution to reduce the number of parameters, which makes the total computation much smaller. However, it also deepens the number of layers of convolution, which may slow down the inference instead of saving the inference time for a massively parallel data processing platform like GPU.



**FIGURE 2.** Model structure diagram,  $C_i$  denotes the feature layer of the backbone network,  $P_i$  denotes the feature layer stacked with up-sampling, and  $T_i$  represents the feature layer fused with multi-scale features. The input image is fed into the backbone feature extraction network after the ARFFE module, and the deepest feature layer is stacked with the upper layer after the channel compression.  $P_0$  is the feature layer stacked with  $C_0-C_3$ . Then the feature fusion and weighting are performed by the downsampling and channel attention modules and finally output to the detection head for detection.

**TABLE 1.** Speed comparison of backbone.

Backbone	Param.	FPS
DarkNet-53	236.32	49.46
MobileNetv2	112.71	62.47
ShuffleNetv2	95.63	66.52
GhostNet	99.75	43.21

We have verified the inference speed with different lightweight networks on the NEU-DET dataset, and the experimental results are shown in Table 1. The experiments show that the actual inference speed on GPU decreases using GhostNet as the feature extraction network, although the parameters are reduced. ShuffleNetv2 is designed with practical inference speed, replacing part of the grouped convolution with ordinary convolution and using concat instead of adding to reduce element-wise operations. ShuffleNetv2 significantly reduces parameters and can obtain faster inference speed during the actual operation. Hence, it is most suitable to be used as a feature extraction network for industrial defect detection models with high requirements for real-time, and its structure is shown in Table 2.

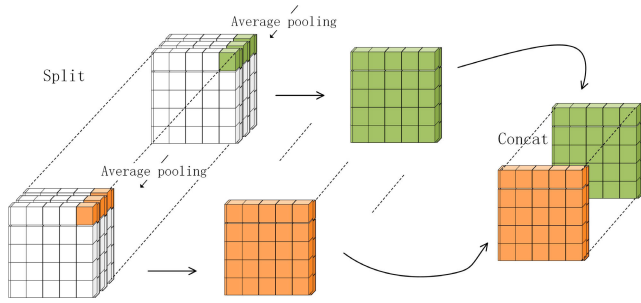
**TABLE 2.** ShuffleNetv2 structure.

Layer	Operator	Output Channel	Stride	Output Size
image	input	3		416x416
Stage0	Conv	128	2	208x208
	Maxpool	256	2	104x104
Stage1	Shufflev2blockx4	256	2	52x52
			1	
Stage2	Shufflev2blockx8	512	2	28x28
			1	
Stage3	Shufflev2blockx4	1024	2	13x13
			1	

### B. LIGHTWEIGHT FEATURE PYRAMID NETWORK

As the network layers deepen, the feature map resolution decreases, and the features of smaller targets disappear. Moreover, more semantic information about the target in the deep network is beneficial in locating the target's position. The original YOLOv3 uses an FPN network structure to fuse multi-scale information to predict objects of different sizes by three layers of feature maps with different resolutions. It is worth noting that the original YOLOv3 does not fuse the feature information of the upper layer when using the feature map of the bottom layer for prediction. Similarly, only the feature information of the bottom layer is fused into the middle layer. PANet uses a top-down and bottom-up bi-directional fusion network to connect all the features of the prediction layer before prediction but at the same time introduces more parameters, making the inference speed





**FIGURE 3.** The feature layer is divided into  $k$  groups by channel in grouped channel compression. Each group is down-sampled by averaging pooling in the channel direction to obtain the feature maps of  $\frac{C}{k}$  channels. Finally, these feature maps are stitched together.

slower. Therefore, this paper proposes a network structure that fuses features more quickly. First, the bottom layer feature map  $C_3$  is downsampled in the channel direction by grouping channel compression (as shown in Figure 3) then upsampled and stacked with  $C_2$  to obtain  $P_2$ , whose number of channels per layer can be calculated by the formula.

$$C_{P_i} = C_{C_i} + \frac{C_{C_{i+1}}}{k}$$

where  $k$  is the number of sub-groups, which is set to 4 in this paper, and  $i$  denotes the feature layer of the feature extraction network,  $i \in \{0, 1, 2\}$ . Similarly, the  $C_1$  and  $C_0$  feature layers are stacked to finally obtain  $P_0$ . The process can be expressed as

$$\begin{aligned} P_2 &= \text{Concat}(C_2, \text{Concat}(\text{CAP}(G_{p_1}), \dots, \text{CAP}(G_{p_n}))) \\ P_1 &= \text{Concat}(C_1, \text{Concat}(\text{CAP}(G_{p_1}), \dots, \text{CAP}(G_{p_n}))) \\ P_0 &= \text{Concat}(C_0, \text{Concat}(\text{CAP}(G_{p_1}), \dots, \text{CAP}(G_{p_n}))) \end{aligned}$$

where  $\text{CAP}$  denotes the channel average pooling, and  $G_{p_n}$  denotes the grouped feature map, the  $n = \frac{C_{P_{i+1}}}{k}$ ,  $k = 4$ , and  $i \in \{0, 1, 2\}$ . It is worth noting that no parameters were introduced during this period. Although the operation of finding the mean value is used in the channel compression, the computation time consumed is much lower than the convolutional computation. A few convolutional layers are used after  $P_0$  to learn how to fuse the stacked feature layers. At the same time, a channel attention mechanism is added for targets at different scales with different sensitivities to individual channel information. Only a small number of parameters are introduced for downsampling, which improves the inference speed of the whole model, and the detection accuracy is also improved because each prediction layer uses a feature map that fuses all the feature layers.

### C. ADAPTIVE RECEPTIVE FIELD FEATURE EXTRACTION

The receptive field is the region's size where each location of the output feature map of each layer of the convolutional network maps to the feature map of the previous layer. A sizeable perceptual field improves the network's performance for the classification task. However, for the target detection task, the

**TABLE 3.** Initialization parameters of our method.

Parameters	Value	Note
Learning rate	1e-2	Initial learning rate
Decay strategy	cosine	
Optimizer	SGD	
Momentum	0.937	
Weight decay	5e-4	
Total epochs	300	
Frozen epochs	50	Freezing backbone
Batch size	8	Set to 12 when frozen

receptive field size should correspond to the anchor set to get better performance. A too-large field of perception will cause the detected area to be too small and ignored as background, resulting in poor detection of small objects. And the too-small field of perception, due to the acquisition of too much local information and causing the loss of global communication, affects the recognition of objects. In the defect detection task, the size setting of the anchor has a large gap due to the multi-scale nature of the defect. This paper proposes an adaptive receptive field feature extraction module (shown in Figure 4), which can be easily inserted into any position of the feature extraction network. The specific process can be expressed as follows.

$$\begin{aligned} P_1 &= \text{Concat}(\text{Conv}_{3 \times 3}, \text{DilateConv}_{3 \times 3}, \text{DilateConv}_{3 \times 3}) \\ P_2 &= \text{Conv}_{1 \times 1}(P_1) \\ P_3 &= P_0 + P_2 \times \text{sigmoid}(\text{RELU}(\text{GAP}(P_2))) \end{aligned}$$

where  $\text{GAP}$  denotes global average pooling, the input  $P_0$  is stacked together after extracting features by  $3 \times 3$  convolution of different receptive fields. Then the number of channels is reduced to the same as the input by  $1 \times 1$  convolution. The convolution of three different receptive fields corresponds to the subsequent prediction on the feature layers of three resolutions.  $P_2$  contains the feature information of different receptive fields. Since the targets of different scales are not equally sensitive to the feature information of different receptive fields contained in the channels, channel attention is used to weigh the feature information. Finally, shortcuts are used to save the information of the original feature map to prevent information loss. The final output  $P_3$  contains the original information and the weighted multi-receptive field information.

## IV. EXPERIMENT

### A. EXPERIMENTAL ENVIRONMENT

The experimental hardware platform is i5-10400F CPU, NVIDIA GeForce RTX3060ti GPU, and we use PyTorch to build our model, PyTorch version 1.11, Cuda version 11.6, experiments are conducted on windows 10 using

TABLE 4. Setting of compared model parameter.

Methods	Learning rate	Decay strategy	Optimizer	Momentum	Weight decay	Epochs	Batch size
YOLOv3	1e-2	cosine	SGD	0.937	5e-4	300	8
YOLOv4	1e-2	cosine	SGD	0.937	5e-4	300	4
ES-Net	1e-2	cosine	SGD	0.937	5e-4	300	-
EfficientDet	1e-2	cosine	SGD	0.937	5e-4	300	8
DCC-CenterNet	1e-3	-	Adam	-	-	300	16
RetinaNet	1e-4	cosine	Adam	0.9	0	300	8
DEA_RetinaNet	1e-5	-	-	-	1e-3	70	-
SSD	2e-3	cosine	SGD	0.937	5e-4	300	16
Faster R-CNN	1e-4	cosine	Adam	0.9	0	300	4
Improved Faster-Rcnn	2e-2	-	-	0.9	1e-4	20	-

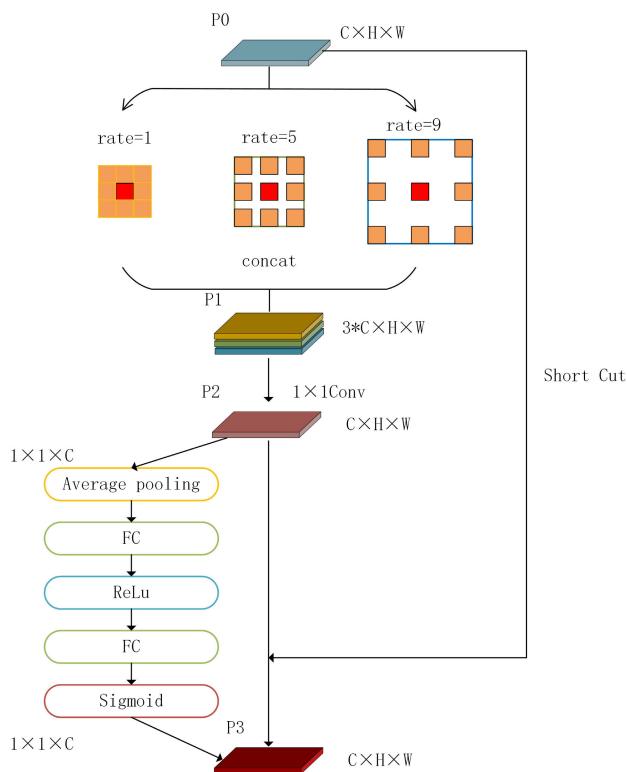


FIGURE 4. Schematic structure of the adaptive perceptual field feature extraction module,  $P_0$  represents the input feature layer, which is stitched together along the channel direction after the convolution of voids with different rates. The channel is compressed by  $1 \times 1$  convolution, and finally, the output  $P_3$  is obtained by adding the channel weighting with the original input through the channel attention module. Output  $P_3$  and input  $P_0$  have the same dimensionality.

pycharm. We use mAP as the evaluation index for model accuracy. Model parameters, FLOPs, and FPS are model speed evaluation indexes. See Table 3 for setting model parameters, and Table 4 for comparing model parameters.

**B. DATASETS**

We use three kinds of data sets, and the data sets are divided by the same ratio. The ratio of the training set and validation set is 9:1. The ratio of the training set plus the validation set

and test set is 8:2, as shown in table 5 for details. The images are randomly enhanced before being input into the network. The data enhancement methods include random flipping and gamut transformation.

- 1) NEU-DET: A steel surface defect detection dataset from Northeastern University, containing six types of defects: rolled-in scale (Rs), patches (Pa), crazing (Cr), pitted surface (Ps), inclusion (In) and scratches (Sc). There are 300 images in each category.
- 2) PCB Defect Dataset: This is a publicly synthesized PCB board defect detection dataset from Peking University, containing six types of defects: Missing hole, Mouse bite, Open circuit, Short Spur, and Spurious coppe.
- 3) GC10-DET: GC10-DET is a dataset of surface defects collected in real industrial scenarios, containing ten types of defects: punch (Pu), weld (Wl), crescent gap (Cg), water spot (Ws), oil spot (Os), silk spot (Ss), inclusions (In), roll pits (Rp), crease (Cr), and waist folding (Wf).

**C. EXPERIMENTS AND ANALYSIS OF RESULTS**

**1) MODEL PERFORMANCE COMPARISON**

To validate the effectiveness of the model, we first compared our model with conventional target detection networks on the NEU-DET dataset, including the one-stage detection networks YOLOv4, EfficientDet [33], RetinaNet, SSD, and the two-stage network Faster-RCNN, and also with other steel surface defect detection models were compared, such as ES-Net, DCC-CenterNet [34], DEA\_RetinaNet [35], Improved Faster-Rcnn [26], and then to verify the generalization performance of the model, we conducted experiments on PCB defect dataset and GC10-DET, all experiments were performed on the same hardware platform, and the experimental results are shown in Table 6-8.

As can be seen from Table 9, our model has the fastest inference speed and the lowest computational complexity. The mAP is not optimal, but compared with the best DCC-CenterNet, the gap is only 0.18%, which is almost

**TABLE 5.** Introduction to three kinds of data set partitioning.

DataSet	Type number	Resolution	Resize	Total number	Train	Val	Test
NEU-DET	6	200×200	416×416	1800	1296	144	360
PCB Defect Dataset	6	3034×1586	640×640	693	498	56	139
GC10-DET	10	2048×1000	512×512	2280	1641	183	456

**TABLE 6.** Detection results of NEU-DET dataset.

Metrics	All	Crazing	Inclusion	Patches	Pitted_surface	Rolled-in_scale	Scratches
Precision(%)	91.38	77.78	96.64	97.83	94.44	85.42	96.15
Recall(%)	62.40	40.64	58.15	76.88	70.33	44.55	83.84
mAP@0.5(%)	79.23	45.11	85.49	94.54	86.31	67.79	96.13

**TABLE 7.** Detection results of PCB defect dataset.

Metrics	All	Missing_hole	Mouse_bite	Open_circuit	Short	Spur	Spurious_copper
Precision(%)	96.25	100	95.38	95.51	97.53	93.15	95.95
Recall(%)	85.02	97.22	86.11	86.73	89.77	68.69	81.61
mAP@0.5(%)	93.31	100	91.69	93.59	98.67	83.44	92.47

negligible in practical application scenarios. At the same time, our parameter amount is only 60.51M, less than half of DCC-CenterNet. Our model inference speed also reaches the fastest 63.24 FPS, it indicate that our model is more valuable for practical applications.

YOLOv3, YOLOv4, and Efficientdet use FPN, PANet, and BiFPN as feature fusion networks, respectively. YOLOv4 improves its performance by 3.9% compared with YOLOv3 using a bidirectional fusion network PANet, which indicates that fusing feature layers with multi-scale features before prediction can improve detection accuracy. However, the number of parameters increases, and the inference speed decreases. Our model uses the proposed LFPN as a feature fusion network and improves by 9.29% compared to the benchmark model YOLOv3 and 5.39% and 9.13% compared to YOLOv4 and Efficientdet, respectively. It confirms that our model structure is superior to the above three models in terms of reference quantity and detection performance.

Compared with other one-stage detection models such as ESNet, RetinaNet, DEA-RetinaNet, and SSD, our model's mAP improves by 0.13%, 19.02%, 0.98%, and 12.16%, respectively. Compared with the two-stage networks Faster R-CNN and Improved Faster-Rcnn, our model has a massive advantage in inference speed, with 4.39 times higher FPS than Faster-Rcnn. In comparison, the detection accuracy is improved by 15.81% and 4.03%, respectively. Our model performs satisfactorily compared to the two-stage network, which is known for its detection accuracy.

To investigate the detection capability of our model for different kinds of defects, we compared each class of defects with other models on the NEU-DET dataset, and the results

are shown in Table 10. It can be seen that the detection of each type of defect is improved compared with the benchmark model. Crazing has relatively fuzzy boundaries causing the detection model challenging to locate the defect location. The original YOLOv3 has an AP of only 28.14% for Crazing, which is almost undetectable. Due to the addition of the ARFFE module, the feature extraction capability of the backbone network is enhanced, the detection capability of such defects as Crazing is improved more significantly, and 16.97% increases the AP. Compared with DEA\_RetinaNet, there is a gap of 15.82%, which is because it adds a difference extraction block between the backbone network and the feature fusion network to reduce the loss of information, and our approach does not have an advantage for defects where the scale varies little. It is not easy to distinguish between boundaries. However, our model has a more flexible receptive field and efficient information fusion capability for defects such as scratches due to the characteristic of excellent scale variation, which makes the detection capability significantly higher than DEA\_RetinaNet. AP improves by 22.08% and is also the best value among all models. The model can easily classify defects such as Patches because of their apparent characteristics. The LFPN network incorporates more feature map layers, allowing the model to better utilize global and local information for defect localization. We also achieve the best detection results for Patches. However, the 8.99% difference between such defects as Rolled-in\_scale and the best-performing DCC-CenterNet leads to a slightly lower final model mAP than DCC-Centernet.

Finally, we did experiments on the PCB defect dataset and GC10-DET to verify the model's generalization ability. The

**TABLE 8.** Detection results of GC10-DET dataset.

Metrics	All	Wf	Pu	Wl	Cg	Ws	Os	Ss	In	Rp	Cr
Precision(%)	76.44	62.07	98.00	97.83	84.62	90.48	91.43	90.00	50.00	0	100
Recall(%)	43.01	75.00	83.05	76.88	61.11	64.41	23.19	29.51	2.7	0	14.29
mAP@0.5(%)	59.78	75.28	94.43	94.54	89.28	79.07	49.28	61.15	14.68	6.16	34.64

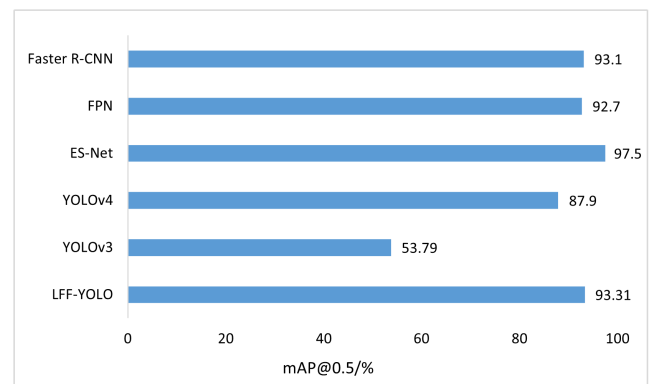
**TABLE 9.** Compare of the detection results and speed with other methods on the NEU-DET datasets.

Methods	Backbone	mAP @0.5/%	FLOPs	Param.	FPS
YOLOv3	Darknet53	69.94	33.09G	236.32M	49.46
YOLOv4	CSP-Darknet53	73.84	30.26G	245.53M	39.92
ES-Net	CSP-Darknet53	79.1	-	147.98M	-
EfficientDet	EfficientNet	70.1	12.56G	199.43M	46.84
DCC-CenterNet	Resnet50	<b>79.41</b>	-	131.24M	-
RetinaNet	Resnet50	60.21	70.01G	139.46M	45.90
DEA_RetinaNet	Resnet50	78.25	-	168.8M	-
SSD	Vgg16	67.07	116.34G	93.16M	59.72
Faster R-CNN	Resnet50	63.42	252.71G	108.64M	14.41
Improved Faster-Rcnn	Resnet50	75.2	-	-	-
Ours	Shufflenetv2	79.23	<b>6.85G</b>	<b>60.51M</b>	<b>63.24</b>

**TABLE 10.** Comparison of the detection results of other methods and the proposed method for each category on the NEU-DET dataset.

Methods	YOLOv3	Efficientdet	ESNet	DEA_RetinaNet	SSD	DCC-CenterNet	Ours
mAP(%)	69.94	70.10	79.1	79.11	67.07	<b>79.41</b>	79.23
Crazing(%)	28.14	45.90	56.0	<b>60.93</b>	38.24	45.72	45.11
Inclusion(%)	74.58	62.00	87.6	82.49	72.14	<b>90.58</b>	85.49
Patches(%)	91.61	83.50	88.3	94.27	87.89	85.05	<b>94.54</b>
Pitted_surface(%)	78.73	85.50	87.4	<b>95.79</b>	72.03	82.49	86.31
Rolled-in_scale(%)	54.06	70.70	60.4	67.16	65.28	<b>76.78</b>	67.79
Scratches(%)	92.52	73.10	94.9	74.05	66.87	95.82	<b>96.13</b>

defects in the PCB defect dataset have the characteristic of small scale. Our model obtained 93.31% mAP, which is only 4.19% lower than the best-performing ES-Net and still has better results than other models. Even though our model is not designed explicitly for small target detection, our model can still make accurate detections for such small defective targets. This is due to the excellent feature fusion capability of LFPN, which makes it possible to effectively fuse global and local information to identify and locate defects when detecting small defect targets accurately. Our model achieves 59.78% mAP on the GC10-DET dataset, which is only 2.15% away from the best-performing DCC-internet, and still has a significant advantage over other mainstream models. In comparing different datasets, our model has a solid competitive detection performance while maintaining the optimal inference speed, which indicates that our model has a powerful generalization capability. At the same time, it achieves the best balance between inference speed and model detection performance.

**FIGURE 5.** Comparison of the detection results of other methods and the proposed method on the PCB defect dataset.

## 2) ABLATION EXPERIMENT

To evaluate the contribution of each module to the model, we set up ablation experiments to assess and analyze the backbone network ShuffleNetv2, the proposed lightweight



TABLE 11. Effect of each module on model accuracy and inference speed.

Baseline	ShuffleNetv2	LFPN	Channel Attention	ARFFE	mAP/%	Param.	FPS
✓	-	-	-	-	69.94	236.32	49.46
✓	✓	-	-	-	70.21	95.63	66.52
✓	-	✓	-	-	74.63	199.26	52.67
✓	✓	✓	-	-	75.58	<b>58.58</b>	<b>67.79</b>
✓	✓	✓	✓	-	76.73	60.46	64.78
✓	✓	✓	✓	✓	<b>79.23</b>	60.51	63.24

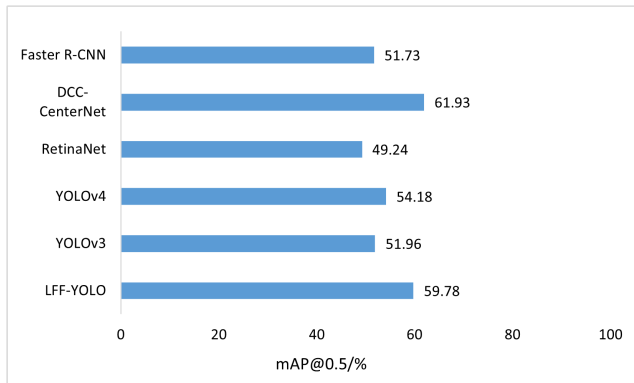


FIGURE 6. Comparison of detection results of other methods and the proposed method on GC10-DET dataset.

feature fusion network LFPN, the adaptive perceptual field feature extraction module ARFFE, and the channel attention module, respectively.

- 1) ShuffleNetv2: replacing the original YOLOv3 backbone feature extraction network Darknet53 with ShuffleNetv2 for comparison experiments, the model parameters decreased from 236.32M to 95.63M, and the FPS increased from the original 49.46 to 66.52 with little change in mAP. This indicates that ShuffleNetv2, as the backbone feature extraction network, can improve the detection speed of the model without sacrificing accuracy.
- 2) LFPN: From the comparison between row 1 and row 3 of the Table 11, we can see that the FPN network in the original YOLOv3 is replaced by the LFPN proposed in this paper, the mAP is increased from 69.94% to 74.63%, and the number of parameters is reduced at the same time. Thus the FPS rose from 49.46 to 52.67. The performance improvement is primarily attributed to the structural design of LFPN, which fuses multi-scale features before down-sampling, and empowers the model to improve detection accuracy while inference is faster. Combined with ShuffleNetV2 as the backbone network, the number of parameters of the model is only 58.58M, while the inference speed reaches the fastest 67.79 FPS.
- 3) Channel Attention module: The channel attention module is introduced in the LFPN downsampling process, and its purpose is to target different scales

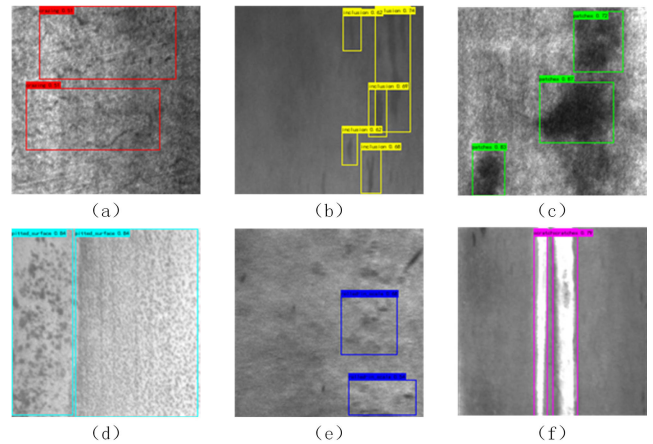


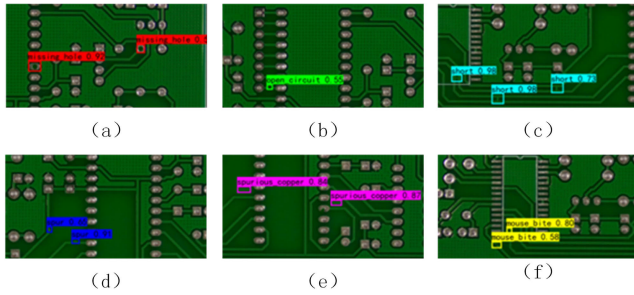
FIGURE 7. Visualization of the detection results of NEU-DET dataset, belonging to the categories: (a) Crazing; (b) Inclusion; (c) Patches; (d) Pitted\_surface; (e) Rolled-in\_scales; (f) Scratches.

of targets with varying sensitivities to each feature channel. Comparing the data in rows 4 and 5 of the table, the results show that adding the channel attention module during downsampling can improve the model performance by introducing a small number of parameters.

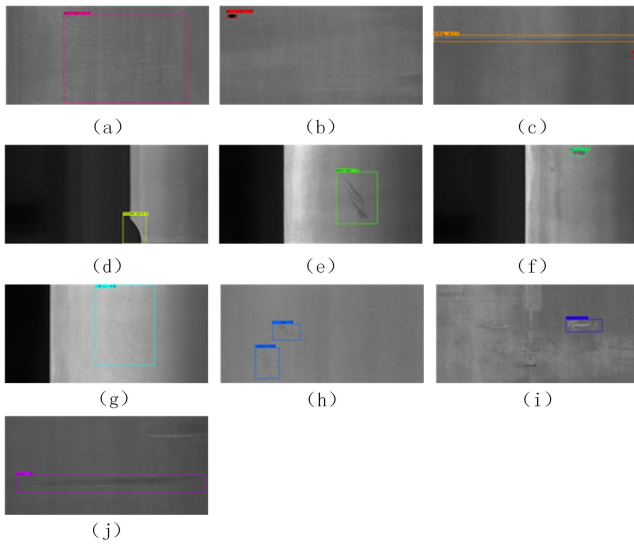
- 4) ARFFE module: ARFFE module is added to the shallow layer of the feature extraction network, as shown in Figure 2. Comparing the last two rows of data in the table, after adding the ARFFE module, the model parameters only increased from 60.46M to 60.51M. In contrast, the mAP rose from 76.73% to 79.23%, which verifies the effectiveness of the ARFFE module. Since the ARFFE module can be inserted into any network position, we put the ARFFE module into the deeper layer of the feature extraction network. Although the model's accuracy is improved, the inference speed of the model is significantly reduced due to the introduction of more parameters in the deeper network with more channels, so the ARFFE module is finally put into the shallow layer of the feature extraction network in this paper.

### 3) VISUALIZATION OF PREDICTION RESULTS

Figure 7 to Figure 9 represent the actual detection result visualization of our model on the NEU-DET dataset, PCB



**FIGURE 8.** Visualization of PCB defect dataset detection results, due to the high resolution of the image and the small prediction frame, only part of the image is captured for display. Categories: (a) Missing\_hole; (b) Open\_circuit; (c) Short; (d) Spur; (e) Spurious\_copper; (f) Mouse\_bite.



**FIGURE 9.** GC10-DET dataset detection visualization, belonging to the categories: (a) Waist\_folding; (b) Punching\_hole; (c) Welding\_line; (d) Crescent\_gap; (e) Water\_spot; (f) Oil\_spot; (g) Silk\_spot; (h) Inclusion; (i) Rolled\_pit; (j) Crease.

defect dataset, and GC10-DET dataset, respectively. Due to the high resolution of PCB defect images, only a portion with defects is captured in the figure as a display. We set a threshold value of 0.5, and the prediction frame is drawn only when the prediction frame score exceeds 0.5. It can be seen that our model can accurately make predictions for various scales of defects in the NEU-DET dataset and GC10-DET dataset, where the GC10-DET dataset has a large image resolution. The scale of defects spans a great deal, from tiny targets such as punching-hole with only a few tens of pixels to those like welding-line that span the entire picture. For these scales span many defects, our model has better detection capability, and LFPN incorporates features of different scales before prediction. The addition of the ARFFE module makes the feature map increase the feature information of multiple sensory fields. In addition, the defect target scale of the PCB defect dataset is microscopic, and our model still has a good detection effect for such small-scale targets.

## V. CONCLUSION

In this paper, we want to solve the industrial defect detection problem and improve the detection accuracy while guaranteeing the inference speed. For this purpose, we designed our model based on YOLOv3. First, we used a faster feature extraction network, ShuffleNet2, to replace the original DarkNet53. To accommodate defects at different scales, we designed the ARFFE module to obtain features for adaptive sensory fields. Then, to improve the fusion efficiency of multi-scale features, we proposed the LFPN network, which enhances the detection accuracy of the network by introducing fewer parameters. The experimental results show that our model reaches 79.23% mAP on the NEU-DET dataset, which is 9.29% higher than the benchmark model. Moreover, we validate the generalization ability of our model on the PCB defect dataset and GC10-DET dataset and reached 93.31% and 59.78% mAP, respectively. Meanwhile, our model has the fastest inference speed, reaching 63.24 FPS on the NEU-DET dataset, which suggests that our model will be beneficial in real industrial application scenarios. Our future work will target model size compression, e.g., using model distillation pruning methods.

## REFERENCES

- [1] S. A. Singh and K. A. Desai, "Automated surface defect detection framework using machine vision and convolutional neural networks," *J. Intell. Manuf.*, 2022, doi: 10.1007/s10845-021-01878-w.
- [2] A. K.-F. Lui, Y.-H. Chan, and M.-F. Leung, "Modelling of destinations for data-driven pedestrian trajectory prediction in public buildings," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 1709–1717.
- [3] A. K.-F. Lui, Y.-H. Chan, and M.-F. Leung, "Modelling of pedestrian movements near an amenity in walkways of public buildings," in *Proc. 8th Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2022, pp. 394–400.
- [4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [8] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.
- [9] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis. Amsterdam, The Netherlands: Springer*, 2016, pp. 21–37.
- [12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
- [13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
- [14] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.
- [15] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

- [16] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.
- [17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
- [18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
- [19] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1580–1589.
- [20] Y. Y. K. Song. (2021). *NEU Surface Defect Database*. [Online]. Available: [http://faculty.neu.edu.cn/songkechen/zh\\_CN/zdylm/263270/list/index.htm](http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list/index.htm)
- [21] Weapon. (2021). *Public Synthetic PCB Dataset*. [Online]. Available: <https://robotics.pkusz.edu.cn/resources/dataset/>
- [22] X. Lv, F. Duan, J.-J. Jiang, X. Fu, and L. Gan, "Deep metallic surface defect detection: The new benchmark and detection network," *Sensors*, vol. 20, no. 6, p. 1562, Mar. 2020.
- [23] P. Prasitmeebon and H. Yau, "Defect detection of particleboards by visual analysis and machine learning," in *Proc. 5th Int. Conf. Eng., Appl. Sci. Technol. (ICEAST)*, Jul. 2019, pp. 1–4.
- [24] C.-F. Chang, J.-L. Wu, K.-J. Chen, and M.-C. Hsu, "A hybrid defect detection method for compact camera lens," *Adv. Mech. Eng.*, vol. 9, no. 8, 2017, Art. no. 1687814017722949.
- [25] F.-L. Wang and B. Zuo, "Detection of surface cutting defect on magnet using Fourier image reconstruction," *J. Central South Univ.*, vol. 23, no. 5, pp. 1123–1131, May 2016.
- [26] W. Zhao, F. Chen, H. Huang, D. Li, and W. Cheng, "A new steel defect detection algorithm based on deep learning," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–13, Mar. 2021.
- [27] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 9, pp. 731–747, Sep. 2018.
- [28] B. Su, H. Chen, P. Chen, G. Bian, K. Liu, and W. Liu, "Deep learning-based solar-cell manufacturing defect detection with complementary attention network," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4084–4095, Jun. 2021.
- [29] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, "A deep learning-based framework for an automated defect detection system for sewer pipes," *Autom. Construct.*, vol. 109, Jan. 2020, Art. no. 102967.
- [30] C. Zhang, C. C. Chang, and M. Jamshidi, "Bridge damage detection using a single-stage detector and field inspection images," 2018, *arXiv:1812.10590*.
- [31] X. Yu, W. Lyu, D. Zhou, C. Wang, and W. Xu, "ES-Net: Efficient scale-aware network for tiny defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.
- [32] R. Wang and C. F. Cheung, "CenterNet-based defect detection for additive manufacturing," *Expert Syst. Appl.*, vol. 188, Feb. 2022, Art. no. 116000.
- [33] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.
- [34] R. Tian and M. Jia, "DCC-CenterNet: A rapid detection method for steel surface defects," *Measurement*, vol. 187, Jan. 2022, Art. no. 110211.
- [35] X. Cheng and J. Yu, "RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.



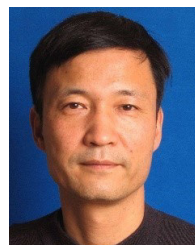
**XIAOHONG QIAN** received the master's degree in transportation engineering from Tongji University. He is a Professor with the Zhejiang University of Science and Technology. His research interests include computer vision, object recognition, and software development.



**XU WANG** received the B.S. degree from the College of Engineering, China West Normal University, Nanchong, China, in 2019. He is currently pursuing the postgraduate degree with the College of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou, China. His research interest includes object detection.



**SHENGYING YANG** (Member, IEEE) received the Ph.D. degree in electronic science and technology from Hangzhou Dianzi University, in 2020. He was a Lecturer with the Zhejiang University of Science and Technology. His research interests include computer vision and deep learning.



**JINGSHENG LEI** received the B.S. degree in mathematics from Shanxi Normal University, in 1987, and the M.S. and Ph.D. degrees in computer science from Xinjiang University, in 2000 and 2003, respectively. He is a Professor with the College of Information and Electronic Engineering, Zhejiang University of Science and Technology. His research interests include machine learning, pattern recognition, and machine vision. He is a member of the Machine Learning Technical Committee, Chinese Association of Artificial Intelligence (CAAI), and Academic Committee of ACM Shanghai Chapter.

...