Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 14, No. 3, 2025

Plastic Parts Surface Defect Detection Algorithm Based on Path-Augmentation Multi-Level Feature Pyramid Network

Mingyang Xu *

The 22nd Research Institute of China Electronics Technology Group Corporation, Xinxiang, Henan, China
*** Corresponding author Email:** 401901966@qq.com

Abstract: Due to the characteristics of plastic parts surface defects such as variable scales and shapes, and a large number of small defect targets, existing algorithms cannot fully extract and utilize defect features. A plastic surface defect detection algorithm based on Path-Augmentation Multi-Level Feature Pyramid Network (PA-MLFPN) is proposed. On the basis of extracting multi-level and multi-scale defect features, to further enhance the representation ability of small defect targets, firstly, dilated convolution is adopted in the encoder of Thinned U-shape Modules (TUM) to fully retain shallow detail features. Meanwhile, a bottom-up feature augmentation path is added to the original TUM to interpolate shallow features into deep layers. Secondly, the Efficient Channel Attention (ECA) module is introduced in the second stage of the Scale-wise Feature Aggregation Module (SFAM) to achieve more reasonable weight allocation across channels. Finally, Focal Loss is used to calculate classification loss, which alleviates the problem of imbalance between positive and negative samples to a certain extent. Embedding the proposed PA-MLFPN into SSD and conducting experiments on the plastic parts surface defect dataset show that the algorithm achieves a mean Average Precision (mAP) of 84.89% and effectively solves the missing detection problem of small defect targets.

Keywords: Plastic Parts Surface Defect Detection; Multi-Level Feature Pyramid Network (MLFPN); Path Augmentation; Dilated Convolution.

1. Introduction

Plastic parts surface defect detection is an indispensable link in plastic production. However, affected by factors such as production equipment and processes, the surfaces of manufactured plastic parts often have various defects including scratches, powder protrusions, and abrasions. These defects not only affect the appearance but also pose potential risks for subsequent use of plastic parts—such as accelerating oxidation to reduce service life and even leading to serious accidents. Therefore, adopting appropriate defect detection technology to improve the production quality of plastic parts is particularly crucial. Traditional surface defect detection is usually performed manually, which lacks standardization due to the influence of subjective experience, working environment, and mental state of inspectors. Meanwhile, manual detection is prone to false detection and missing detection when inspecting high-speed moving workpieces or those with small defects.

With the development of deep learning, an increasing number of object detection algorithms based on Convolutional Neural Networks (CNNs) have been proposed. These algorithms are mainly divided into two categories: two-stage object detection algorithms and one-stage object detection algorithms. Two-stage algorithms, represented by Fast R-CNN[1], Faster R-CNN[2], and Mask R-CNN[3], are based on region proposal. They first generate region proposals on feature maps extracted by CNNs to initially distinguish foreground from background and determine approximate positions, then perform fine-grained classification and bounding box regression. Their advantage lies in high detection accuracy. One-stage algorithms, such as YOLO[4], SSD[5], YOLO9000[6], and YOLOv3[7], are regression-based. They directly conduct classification and regression on CNN-extracted feature maps, featuring fast detection speed.

The aforementioned CNN-based object detection algorithms were initially proposed for natural image detection tasks. Subsequently, their application in surface defect detection of different materials has become widespread. In 2017, Cha et al.[8] first replaced the backbone network of Faster R-CNN with ZF-net and applied it to bridge surface defect detection, achieving an mAP of 0.878. In 2018, Chang Haitao et al.[9] proposed a defect detection method based on Faster R-CNN. Instead of generating 9 types of anchor boxes with 3 scales and 3 ratios, they selected 42 types of anchor windows according to the aspect ratio and size of rectangles, achieving a detection accuracy of up to 96% with an average detection time of 86 ms per image. He et al.[10] proposed a plastic strip surface defect detection network based on Faster R-CNN, which improves by combining multi-level feature maps in the backbone into a multi-scale feature map. On the plastic defect detection dataset NEU-DET, their method achieved an mAP of 82.3% with ResNet50 as the backbone. Zhang Lei et al.[11] used YOLOv3 for aluminum profile surface defect detection, fusing original images with preprocessed images, and optimizing network feature extraction and generalization capabilities through K-means clustering and parameter tuning.

Due to the limitations of CNN structures and the characteristics of plastic parts surface defects, we identify two key issues in current CNN-based plastic parts surface defect detection: (1) The random generation of plastic parts surface defects results in variable defect shapes and scales. For example, two different types of defects may have similar sizes but significantly different complexities, or defects of the same

type may vary greatly in scale. However, in traditional CNN models, feature layers used to detect objects within specific size ranges are mainly composed of single-level or adjacent two-level feature layers for final detection, leading to poor performance. (2) In practical plastic parts surface defect detection, besides variable shapes and scales, there are often a large number of small defect targets. In CNNs, shallow feature layers have higher resolution and contain more positional and detail information beneficial for small target detection. Deep feature layers have stronger semantic information but low resolution, resulting in poor perception of small targets. Therefore, effectively utilizing shallow features is crucial for detecting small defect targets.

To address these issues, we initially select the Multi-Level Feature Pyramid Network (MLFPN)[12] to handle defects with variable scales and complexities by extracting multi-level and multi-scale features. Based on MLFPN, we propose improvements for plastic parts surface defect detection, namely the Path-Augmentation Multi-Level Feature Pyramid Network (PA-MLFPN), to further utilize shallow features and improve the detection accuracy of small defect targets.

2. Related Work

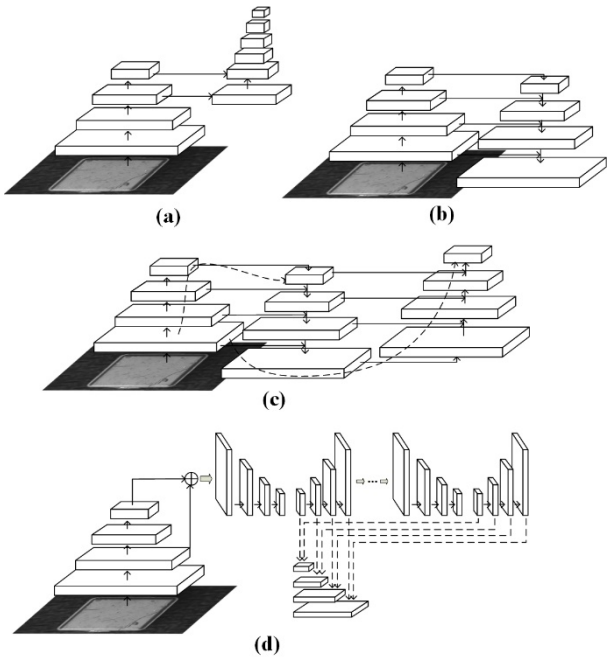


Fig 1. Several forms of feature pyramids

Variable target scales are a major factor affecting detection accuracy. In recent years, numerous researchers have proposed various forms of feature pyramids to address this issue. As shown in Figure 1(a), SSD constructs a feature pyramid using two feature layers from the backbone network and four additional layers obtained through further convolution. However, this approach directly uses feature layers from different levels for prediction without fully fusing multi-scale features. Figure 1(b) shows that FPN[13] adds a new top-down path to compensate for the lack of semantic information in shallow layers. Nevertheless, FPN only performs top-down feature fusion and only fuses adjacent two levels, failing to fully utilize information from all layers. As illustrated in Figure 1(c), PA Net[14] addresses this limitation by constructing a feature augmentation path (dashed line) in FPN, proposing a bottom-up secondary fusion model to further utilize shallow features, strengthen the feature pyramid, and verify the effectiveness of bidirectional fusion. MLFPN proposed in the M2Det algorithm (Figure 1(d)) aims to build a more effective multi-level feature pyramid for detecting targets with variable scales. By stacking multiple U-shaped encoding-decoding structures at different levels and fusing feature layers of the same scale extracted by these U-shaped structures, MLFPN generates a multi-level feature pyramid to extract multi-level and multi-scale features.

In addition to variable scales, plastic parts surface defects are characterized by a large number of small targets. To solve the small defect detection problem, Cheng Jingyi et al.[15] fused the 11th shallow feature with deep network features based on the YOLOv3 structure, generating a new feature map with a scale of 104×104 to extract more small defect features. Li Weigang et al.[16] adjusted the YOLOv3 network structure, fusing shallow and deep features to obtain large-scale feature layers for prediction, thereby improving the detection accuracy of tiny defects. Wang Haiyun et al.[17] improved the FPN structure in Mask R-CNN by adding a bottom-up path to fully fuse feature maps at all stages, obtaining multi-scale feature maps with stronger semantic and positional information. These methods adopt a bottom-up fusion approach, i.e., interpolating detail and positional information from shallow features into deep layers, indicating that fusing multi-scale features—especially shallow features—is an important factor in improving the detection accuracy of small defect targets.

Therefore, this paper improves MLFPN to propose PA-MLFPN, which enhances the utilization of shallow features to better perform plastic parts surface defect detection. The main improvements are as follows:

(1) Introduce dilated convolution in the encoder of each Thinned U-shape Module (TUM) to fully retain defect target information by expanding the local receptive field. Additionally, add a bottom-up feature augmentation path to the original U-shaped structure of TUM to interpolate shallow features into deep layers, further utilizing shallow features in each TUM.

(2) Introduce the Efficient Channel Attention (ECA) module[18] in the second stage of the Scale-wise Feature Aggregation Module (SFAM) to enable more reasonable weight allocation across channels.

(3) Use Focal Loss[19] to calculate classification loss in the loss function, alleviating the problem of imbalanced positive and negative samples in plastic parts surface defect detection.

3. Proposed Work

In this paper, the proposed PA-MLFPN embedded in SSD is used for surface defect detection of plastic parts. The network model is shown in Figure 2. Select VGG16 as the backbone feature extraction network, and incorporate conv4 in VGG16_3 and conv5_3 is sent into PA-MLFPN. PA-MLFPN consists of three Modules: the Feature Fusion Module (FFM), the improved TUM, and the improved Scale-wise Feature Aggregation Module (SFAM). In PA-MLFPN, conv4 is first processed through FFMv1_3 and conv5_The initial feature fusion is carried out to obtain the basic feature layer (Base feature) for the subsequent series of further feature fusions. Then, multiple levels of TUM and FFMv2 are alternately stacked. TUM performs further feature extraction and outputs feature maps of six scales at different levels. FFMv2 fuses the basic feature layer and the maximum feature map output by the previous level TUM, and finally sends the fused feature map to the next level TUM. By stacking

multiple levels of TUM, feature pyramids at multiple levels from shallow to deep were output. Finally, through the first stage of SFAM, the feature layers of the same size in multiple feature pyramids are aggregated, and then in the second stage, the ECA module is used to allocate the weights on the channels. Finally, dense bounding boxes and category scores are generated based on the learned features. Since the number of candidate prediction boxes is relatively large, the candidate prediction boxes need to be sorted by score and the non-maximum Suppression (NMS) operation is performed to obtain the final prediction box. The following will provide a detailed explanation of each module of PA-MLFPN.

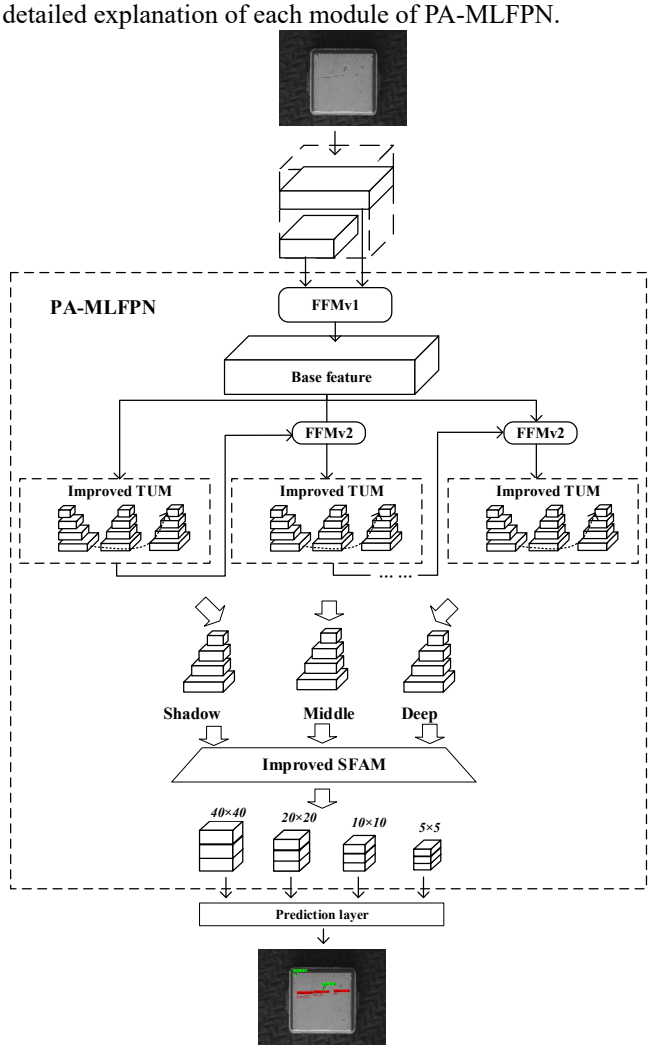


Fig 2. The network model in this article

3.1. Feature Fusion Modules (FFMs)

FFM consists of two parts: FFMv1 and FFMv2. Its purpose is to fuse features at different levels, thereby constructing the final multi-level feature pyramid. The structural details of FFMv1 and FFMv2 are respectively shown in Figures 3 (a) and (b).

FFMv1 extracts conv4 from the backbone feature extraction network VGG_3 and conv5_Two feature layers are used as input. And for conv5_The feature layer performs a convolution with 512 channels, a convolution kernel size of 1×1, and a step size of 1 to adjust the number of channels, and then increases the feature size through upsampling operation. At the same time, conv4_The feature layer performs a convolution adjustment with 256 channels, a convolution kernel size of 3×3, and a step size of 1 to adjust the number of channels. Finally, the two adjusted feature layers are concatenated as the basic feature layers that have undergone initial fusion.

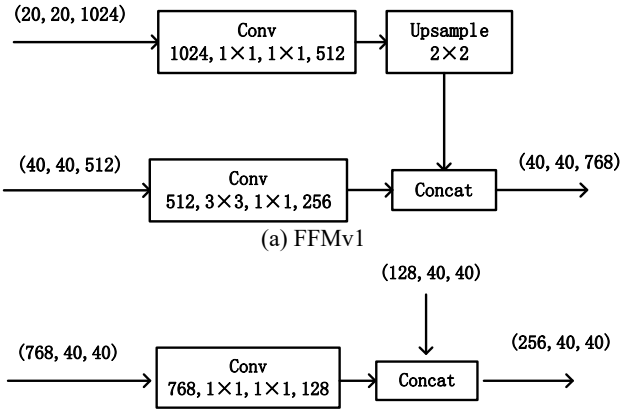


Fig 3. FFMs structure diagram

To further enhance the feature extraction capability of the network, FFMv2 fuses the basic feature layer and the shallowest layer in the previous TUM, that is, the maximum output feature map, as input. FFMv2 adjusts the number of channels by performing a convolution with 128 channels, a convolution kernel size of 1×1, and a step size of 1 on the basic feature layer obtained in FFMv1, and concats it with the maximum output feature layer of the upper-level TUM as the input of the deeper level TUM.

3.2. Improved TUMs

The original TUM structure is shown in Figure 4. Each level of the TUM module is a U-shaped network structure, which is divided into an encoder part and a decoder part.

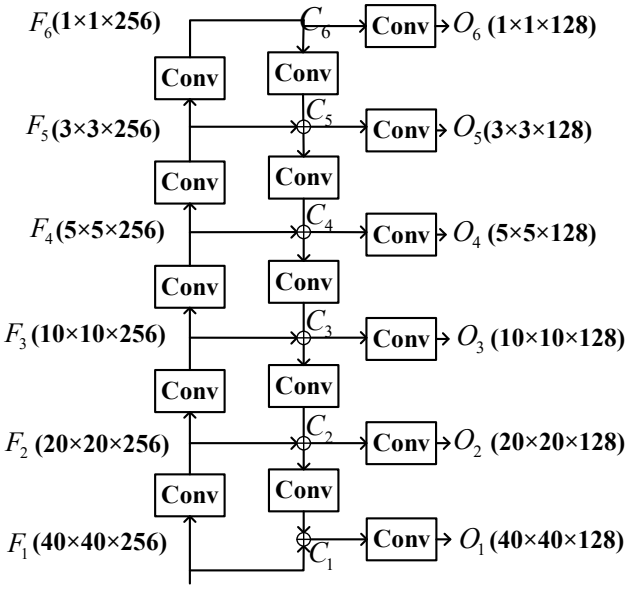


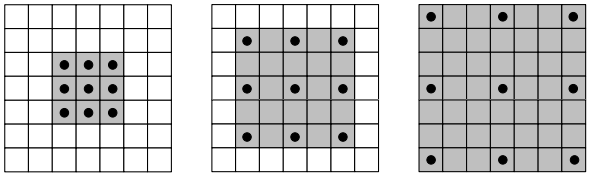
Fig 4. Original TUM structure diagram

The output of the six intermediate feature layers in the decoder of each level of TUM forms the feature pyramid of the current level. That is, the front TUM produces the shallow feature pyramid, the middle TUM generates the middle feature pyramid, and the rear TUM generates the deep feature pyramid. In each original level of TUM, the output of the six feature layers is:

$$
\begin{aligned}
O_6 &= Conv(F_6) \\
O_5 &= Conv(Conv(C_6) + F_5) \\
O_4 &= Conv(Conv(C_5) + F_4) \\
O_3 &= Conv(Conv(C_4) + F_3) \\
O_2 &= Conv(Conv(C_3) + F_2) \\
O_1 &= Conv(Conv(C_2) + F_1)
\end{aligned} \tag{1}
$$

In the formula, the six intermediate feature layers of the encoder are defined as {F1, F2, F3, F4, F5, F6}, and the six intermediate feature layers of the decoder are defined as {C1, C2, C3, C4, C5, C6}, where C6 is equal to F6. The output of the six feature layers from shallow to deep in each stage of TUM is defined as {O1, O2, O3, O4 O5, O6}. The outer Conv is a 1×1 convolution with a step size of 1 and 128 channels for adjusting the final output channel, and the inner Conv is a 3×3 convolution with a step size of 1 and 256 channels for adjusting the size of the feature layer. However, the top-down fusion approach in TUM, similar to that in FPN, merely compensates for the insufficiency of semantic information of shallow features at this level of TUM. It can be seen that the information of O6 in TUM is only provided by F6 that has undergone multiple convolution processes. Therefore, there is still less shallow feature information in the output of the deeper layers in TUM. This will also lead to the lack of shallow detail position information in the multiple feature pyramids from shallow to deep in the final multi-level TUM output, making it impossible to detect small defect targets well. So below we propose an improved TUM for this issue.

Firstly, due to the fact that the TUM encoder part uses a series of downsampling operations to reduce the size of the feature map, this process inevitably causes a loss in accuracy. Therefore, we introduce dilated convolution to increase the receptive field while keeping the size of the feature map unchanged. As shown in Figure 5, with the same 3×3 convolution kernel, dilated convolution can expand the receptive field without increasing the computational load. The gray area represents the receptive field after convolution. In the downsampling part of each level of TUM, dilated convolution with different dilation rates is used instead of traditional convolution. Multi-scale information is obtained by obtaining receptive fields of different sizes, while avoiding information loss during the downsampling process to retain more shallow defect target information.



(a) dilation rate = 1 **(b) dilation rate = 2** **(c) dilation rate = 3**

Fig 5. Schematic diagram of dilated convolution

On this basis, we introduce the idea of bottom-up path augmentation in each level of TUM. On the basis of the original U-shaped structure, we add an information transmission path from the shallow layer to the deep layer to interpolate the detailed position information of the shallow layer features to the deep layer. The schematic diagram of feature fusion is shown in Figure 6. Ti undergoes a convolution with a kernel size of 3×3 and a step size of 2, reducing the size of the feature map to half of its original size. Then, it is added element-wise with Ci+1, and the resulting convolution is further processed with a kernel size of 3×3 and a step size of 1 to obtain Ti+1.

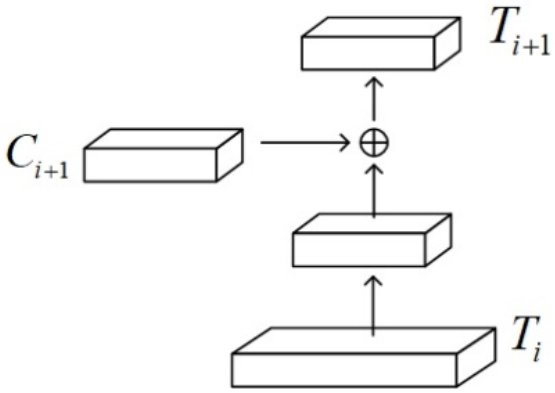


Fig 6. Schematic diagram of feature fusion

Combining the above two points, the improved TUM structure is shown in Figure 7. On the basis of using dilated convolution instead of traditional convolution in the encoder part, a path that can transfer shallow information from bottom to top is added to the original U-shaped structure of TUM. The improved TUM has undergone a secondary fusion from bottom to top in the black dashed box.

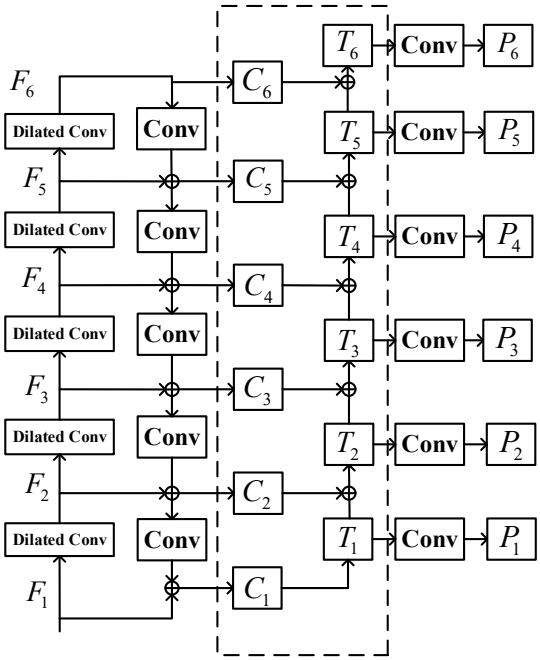


Fig 7. Improved TUM structure diagram

The combined calculation formula for the multi-level and multi-scale features output by the improved multi-level TUM is:

$$
[x_1^l, x_2^l, \mathrm{K}\ , x_i^l] = \begin{cases} T_l(X_{base}), & l = 1 \\ T_l(F(X_{base}, x_i^{l-1})) & l = 2\mathrm{K}\ L \end{cases} \tag{2}
$$

Among them, Xbase represents the basic features, xli represents the feature with the i-th scale in the l-th TUM, L represents the number of TUMs, T1 represents the operation performed by the l-th TUM, and F represents the operation performed by FFMv2. By strengthening the utilization of high-resolution shallow feature information in each level of TUM respectively, the multi-level and multi-scale features

further combined with shallow features are obtained through stacking the outputs of multi-level TUMs, thereby improving the detection accuracy of small defect targets. As shown in Figure 8, it is a comparison of the detection effects before and after the improvement of TUM. After the improvement, relatively ideal detection results are achieved for the originally missed small scratches and convex powder defect targets.

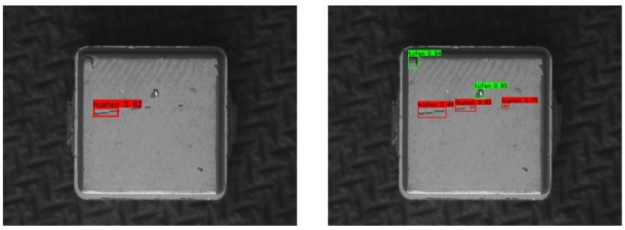


Fig 8. Comparison of the detection results on the test set before and after the improvement of TUM

3.3. Improved Scale-based Feature Aggregation Module (Improved SFAM)

The original TUM structure is shown in Figure 4. Each level of the TUM module is a U-shaped network structure, which is divided into an encoder part and a decoder part.

The SFAM module aggregates multi-level and multi-scale features produced by multi-level TUM to construct a multi-level feature pyramid ultimately used for prediction. As shown in Figure 9, in the first stage, the SFAM module aggregates the feature layers of the same size from multiple feature pyramids from shallow to deep. The aggregated feature pyramid can be represented as:

$$X = [X_1, X_2, \mathrm{K}\ X_i] \tag{3}$$

$$X_i = Concat(x_i^1, x_i^2, \mathrm{K}\ , x_i^l) \in R^{W_i * H_i} \tag{4}$$

Among them, Xi is the feature of the i-th scale, l is the number of layers, Concat is the aggregation operation, and is the spatial size. That is, after aggregating features of the same size, each feature layer at each level contains a multi-level feature pyramid composed of features from different levels.

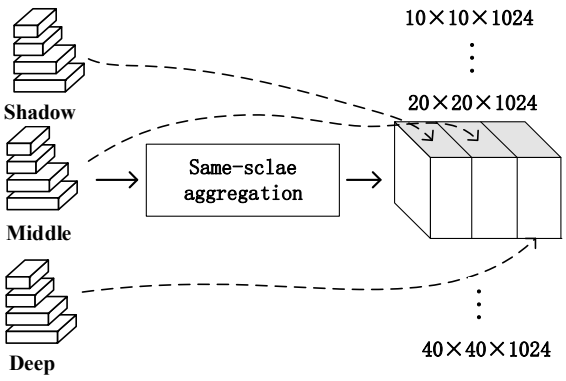


Fig 9. The feature aggregation module in SFAM

However, on this basis, since each feature layer after the first-stage aggregation is only formed by concatenating feature layers of the same size from different levels in the first stage, there is no integration on the channel. Therefore, in the second stage of SFAM, we introduce the ECA channel attention module, and the improved SFAM is shown in Figure 10. In the original SE module, two fully connected layers are used to calculate channel weights, and dimensionality reduction is employed to control the complexity of the model. However, the dimensionality reduction operation will also damage the mapping relationship between channels and their weights. Therefore, the ECA module is introduced, which uses a 1D convolution with size k to replace the two fully connected layers, where k is adaptively determined by a function of the number of channels C. This not only avoids the dimensionality reduction operation but also enables the interaction of local cross-channel information through 1D convolution with an adaptive size.

Specifically, for the feature layers initially aggregated in the first stage of SFAM, global average pooling is first performed to obtain the weight of each channel. If there are C channels, a total of C channel weights is obtained. Then, a 1D convolution with size k is used to achieve local cross-channel information interaction to obtain the weights, where the kernel size k needs to be adaptively determined:

$$k = \psi(C) = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{5}$$

$\left| t \right|_{odd}$ means taking the nearest odd number, and b is set to 1 and 2 respectively in the experiment. Finally, the output from the previous step is passed through the Sigmoid function to obtain C values between 0 and 1, which correspond to the weights of the original channels respectively. These weights are then multiplied by the input features for weighting, and the result is output.

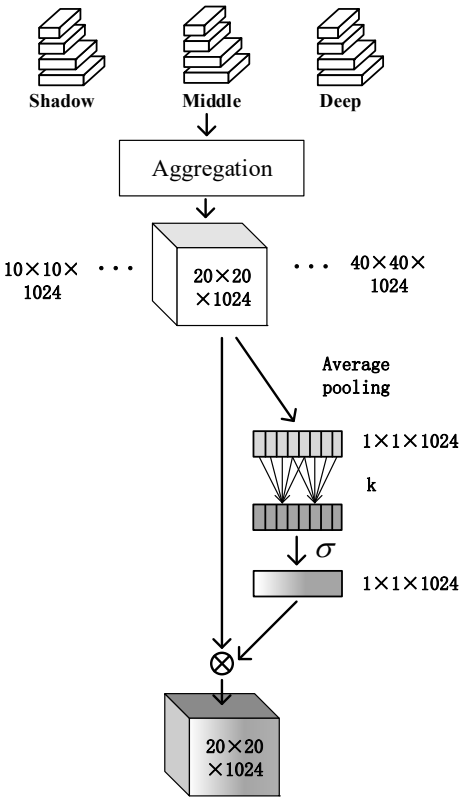


Fig 10. Improved SFAM

3.4. Loss Function

This paper proposes that when PA-MLFPN is embedded into SSD for detection, dense prior boxes are generated in each grid of the entire image during bounding box prediction. This results in the majority of candidate boxes during training being background classes, i.e., easily distinguishable negative samples, especially in plastic surface defect images where

there are few defect targets. A large number of easy negative samples from background classes do not provide useful learning information, which not only leads to a large number of ineffective training processes but also causes their generated loss to overwhelm the loss of a small number of positive samples. That is, the gradients of easy negative samples dominate and lead to a decline in model performance.

Therefore, this paper adds Focal loss to solve the above problems. During training, Focal loss is used to calculate the classification loss, and Smooth L1 is used to calculate the regression loss. Finally, the loss function adopted in this paper is a combination of Focal loss and Smooth L1:

$$L = L_{fl} + L_{SL1} \tag{6}$$

The calculation formula of the classification loss Focal loss is as follows:

$$L_{fl} = \begin{cases} -\alpha(1-y')^{\gamma} \lg y' & y = 1 \\ -(1-\alpha) y'^{\gamma} \lg(1-y^{\gamma}) & y = 0 \end{cases} \tag{7}$$

Among them, y is the predicted output, y is the label of the real sample, α is the weight of positive and negative samples, and γ is the weight of easily classified samples and hard-to-classify samples.

The calculation formula of regression loss Smooth L1 is as follows:

$$L_{SL1} = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{8}$$

Where x is the difference between the predicted bounding box and the ground truth bounding box.

4. Experiment

4.1. Experimental Platform

The experimental operating environment is the Windows 10 system, with an Intel(R) Core(TM) i7-8700 CPU running at a frequency of 3.70 GHz, 16 GB of memory, and a GeForce GTX 1080Ti GPU. Based on the above hardware, training was conducted on PyCharm using Python's Keras framework, and third-party libraries were installed to support model training.

4.2. Dataset

In the experiment, a dataset of plastic part surface defect images collected by high-definition cameras in the factory was used. The plastic part surface defect images are shown in Figure 11: the yellow solid line frame indicates scratch defects, the red solid line frame indicates convex powder defects, and the blue solid line frame indicates abrasion defects.

A total of 210 images for plastic part surface defect detection were collected this time. First, manual annotation was performed using the labelImg image annotation software. Then, data augmentation was implemented on these 210 images through operations such as horizontal flipping, vertical flipping, brightness enhancement, introduction of Gaussian noise, and sharpening, resulting in a total of 2310 images in the dataset. The augmented dataset was randomly divided into (training set + validation set) and test set in a ratio of 9:1, and the training set and validation set were also divided in a ratio of 9:1, specifically 1871 images in the training set, 208 images in the validation set, and 231 images in the test set.

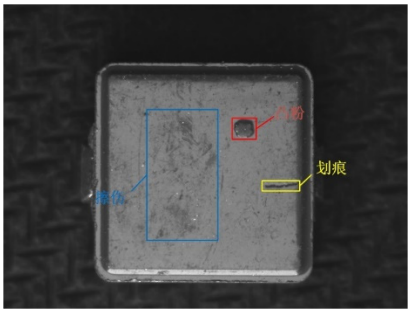


Fig 11. Surface defect images

4.3. Model Training and Evaluation Metrics

This paper adopts the transfer learning method to train the network. First, the weight file is obtained by pre-training on the VOC dataset, and then fine-tuning is performed on the steel surface defect dataset. The number of cyclic iteration steps is set to 100. Initially, the batch size is set to 32, and the learning rate is initialized to 5e-4. When the number of iteration steps reaches 50, the batch size is reset to 16 and the learning rate is set to 1e-4. In addition, the early stopping method is used during training to avoid overfitting caused by continued training. The validation loss is calculated in each iteration. When the validation loss reaches a local optimum, the iteration continues for another 6 times, and the training stops if the model no longer converges.

To evaluate the performance of the algorithm proposed in this paper, relevant evaluation metrics, namely Precision (P) and Recall (R), are used for model evaluation. Their calculation formulas are shown in equations (9) and (10) respectively. Among them, TP, TN, and FP represent the number of correctly identified defects, the number of correctly identified background regions, and the number of background regions mistakenly identified as defects, respectively.

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$R = \frac{TP}{TP + FN} \tag{10}$$

Average Precision (AP) is used to evaluate the model's performance in detecting various types of defects on the test set, as shown in Equation (11). Figure 12 presents the PR curves plotted with P as the horizontal axis and R as the vertical axis, based on the experiments of the algorithm proposed in this paper on the steel surface defect dataset. The area enclosed by the PR curve and the horizontal and vertical coordinate axes is the AP for that type of defect.

The detection results of multi-class defects are evaluated using the mean Average Precision (mAP), and the mAP in this paper is shown in Equation (12).

$$AP = \int_0^1 P(R)dR \tag{11}$$

$$mAP = \frac{AP_{划痕} + AP_{凸粉} + AP_{擦伤}}{3} \tag{12}$$

In addition, the detection speed of the algorithm is measured by the number of images processed per second, i.e., Frames Per Second (FPS).

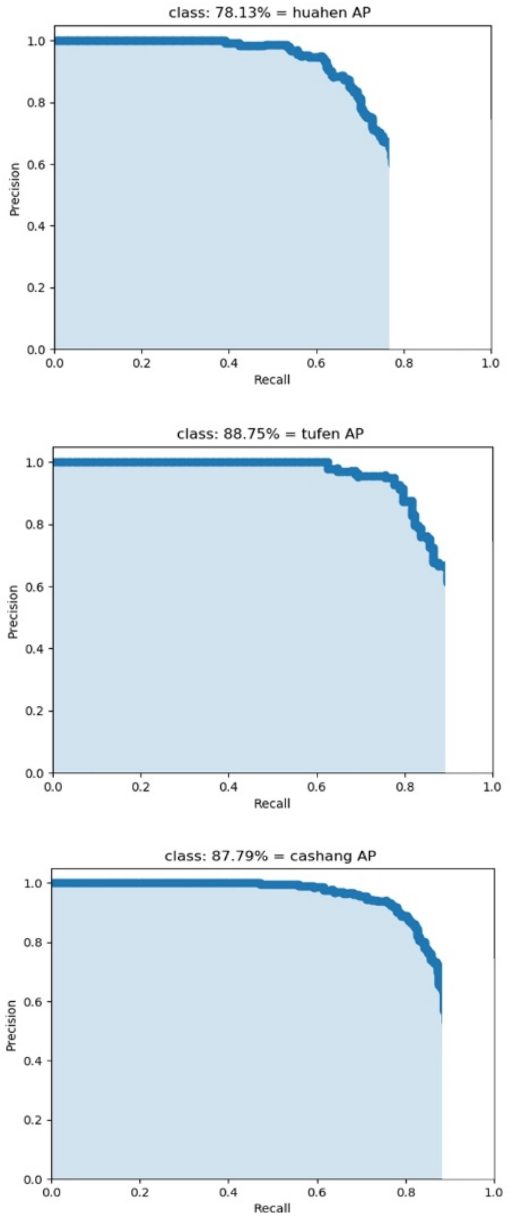


Fig 12. P-R curves for each defect type

4.4. Experimental Results and Analysis

The algorithm in this paper further extracts deeper-level multi-scale features by stacking multiple levels of TUMs. The number of stacks was set to 4 or 8 by the authors in the original MLFPN. Before conducting the comparative experiments, we performed comparative experiments on this hyperparameter using 4 and 8 respectively on the plastic surface defect dataset, and the experimental results are shown in Table 1. It was found that when the number of TUMs is 4, the mAP is higher and the detection speed is faster. Therefore, subsequent experiments were carried out on the basis of stacking 4 TUM modules.

Table 1. Comparison of experimental results on the number of stacked TUMs

Number of TUMs	mAP/%	FPS
4	80.03	25.42
8	78.57	24.89

To analyze the impact of each improvement method in our proposed algorithm on the detection results of surface defects in plastic parts, we added the improvements one by one and conducted comparative analyses to verify the impact of each improved part. The experimental results are compared in Table 2. The PA-MLFPN proposed in this paper has three improvements compared to MLFPN:

(1) In the encoder part of each level of TUM, dilated convolution is used instead of traditional convolution, and a bottom-up feature enhancement path is added to the original TUM structure. It can be seen that after improving the TUM part, the mAP increased by 3.08%, especially the AP of the two defect types with more small defects, namely scratches and convex powder, increased by 5.66% and 2.79% respectively, indicating that the detailed position information provided by the shallow features in each level of TUM has been further utilized.

(2) The ECA module is introduced in the second stage of SFAM for weight allocation on channels. After introducing the ECA module, the AP of the three defect types all have slight improvements, and the mAP increased by 0.92%, indicating that it completes the weight allocation on channels more reasonably compared to the SE module.

(3) After using Focal loss for classification loss in the loss function part, the mAP increased by 0.86%.

Table 2. Comparison of experimental results improved based on MLFPN

Method	AP/%			mAP/%
	scratch	convex powder	abrasion	
MLFPN(M2Det)	72.19	83.56	84.34	80.03
+ dialated conv +path augmentation	77.85	86.35	85.12	83.11
+ECA module	77.93	88.81	85.34	84.03
The algorithm in this paper	78.13	88.75	87.79	84.89

To further verify the comprehensive detection performance of the algorithm in this paper, we conducted experiments on other commonly used target detection algorithms, such as the two-stage target detection algorithm Faster R-CNN, as well as the single-stage target detection algorithms YOLOv5 and SSD, on the plastic surface defect dataset. Table 3 shows a comparison of the experimental results of different algorithms, and the comparison of detection results before and after improvement is shown in Figure 13.

Experimental results show that the algorithm proposed in this paper achieves a greater improvement in AP compared to other algorithms for two types of smaller defects, namely scratches and convex powder, in the plastic surface defect dataset, and there is also a significant improvement in the overall mAP. From the comparison chart of detection results, it can be seen that the algorithm proposed in this paper can

accurately locate and identify various types of defects, while the original algorithm has missed detection phenomena for smaller scratches and convex powder defects. The drawback is that although the detection speed is faster than that of Faster R-CNN, it is slightly lower than that of similar one-stage object detection algorithms. However, it still basically meets the real-time requirements of actual plastic surface defect detection tasks. Therefore, it is proven that the algorithm proposed in this paper meets the needs of plastic surface defect detection tasks, and its overall performance is superior to other similar algorithms in the table, making it more suitable for solving the problem of plastic surface defect detection.

Table 3. Comparison of experimental results of different algorithms

Method	AP/%			mAP/%
	scratch	convex powder	abrasion	
YOLOv5	73.24	80.32	83.35	78.97
Faster R-CNN	72.18	83.26	69.79	75.08
SSD	75.85	79.98	72.35	76.06
MLFPN(M2Det)	72.19	83.56	84.34	80.03
The algorithm in this paper	78.13	88.75	87.79	84.89

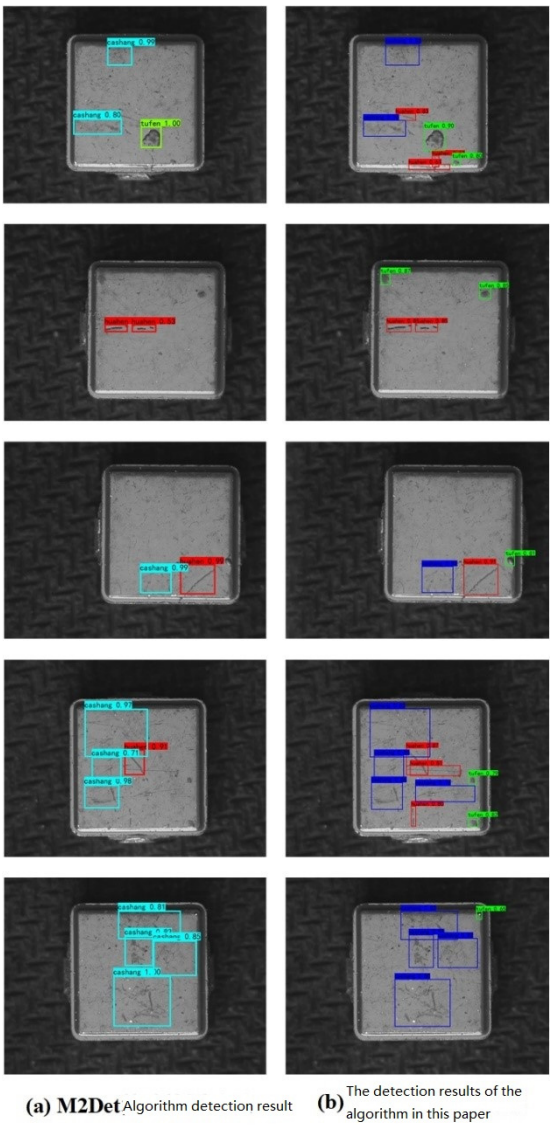


Fig 13. Comparison chart of detection results

5. Summary

This paper first presents two issues in the actual task of detecting surface defects in plastic parts: the varying scales and shapes of defects, and the large number of small defect targets based on this. To address these two issues, we propose PA-MLFPN, which enhances the representation ability of small defect targets by further utilizing shallow features on the basis of extracting multi-level and multi-scale defect features, and embeds the proposed PA-MLFPN into SSD for detection. Experimental results on the plastic part surface defect dataset show that the mAP of the proposed algorithm in this paper is 84.89%, which is 4.86% higher than the original MLFPN, and also higher than the current mainstream target detection algorithms, meeting the accuracy requirements for the task of detecting surface defects in plastic parts. However, since defect detection tasks often have high real-time detection requirements, the detection speed of this model still needs to be improved. In the next step, we will continue to optimize the network model, reduce the model parameters, and try to actually deploy and apply it to the workpiece production line.

References

[1] Girshick R. Fast R-CNN[C]//IEEE Internetional Conference on Computer Vision (ICCV),2015:1440-1448.

[2] Ren S Q, He K M,Girshick R,et al.Faster R-CNN:Towards Real-Time Object Detection with Region Proposal Networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017,39(6):1137-1149.

[3] He K M, Gkioxari G,Dollár P, et al. Mask R-CNN[C]//Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 2961-2969.

[4] REDMON J,DIVVAL A S,GIRSHICK R,et al.You Only Look Once:Unified,Real-Time Object Detection[C]//Computer Vision &Pattern Recognition.IEEE,2016:779-788.

[5] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single Shot MultiBox Detector [J]. 2016:21-37.

[6] REDMON J, FARHADI A. YOLO9000: Better,Faster, Stronger[C]// IEEE Conference on Computer Vision&Pattern Recognition. IEEE,2017:6517-6525.

[7] REDMON J, FARHADI A. YOLOv3:an incremental improvement [J]. arXiv:1804.02767,2018.

[8] Cha Y J, Choi W, Suh G, Mahmoudkhani S, Buyukozturk O. Autonomous structural visual inspection using region based deep learning for detecting multiple damage types. Computer-Aided Civil and Infrastructure Engineering, 2018,33(9):731-747.

[9] Chang, H. T., Gou, J. N., Li, X. M. The application of Faster R-CNN in defect detection of industrial CT images. Journal of Image and Graphics, 23(07), 1061-1071.

[10] He Y, Song K, Meng Q, Yan Y. An End-to-end Plastic Surface Defect Detection Approach via Fusing Multiple Hierarchical Features. IEEE Transactions on Instrumentation, 2020,69(4): 1493−1504.

[11] Zhang, L., Lang, X. L., Wang, L. The surface defect detection of aluminum profiles based on image fusion and YOLOv3. Computer and Modernization, (11), 8-15.

[12] Zhao Q, Sheng T, Wang Y, et al.M2det:A single-shot object detector based on multi-level feature pyramid network[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2019,33:9259-9266.

[13] Lin T Y, Dollár P, Girshick R, et al. Feature Pyramid Networks for Object Detection[J].2016:936-944.

[14] Liu S, Qi L, Qin H, et al. Path Aggregation Network for Instance Segmentation[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA: IEEE,2018:8759-8768.

[15] Cheng, J. Y., Duan, X. H. Zhu, W. The research on metal surface defect detection based on the improved YOLOv3. Computer Engineering and Applications, 1-9.

[16] Chen, K. Xu, X. H. The application research on surface defect detection of aluminum profiles based on the improved Faster RCNN. Journal of China Jiliang University, 31(02), 240-246.

[17] Wang, H. Y., Wang, J. P., Zhang, G., Ouyang, X. Luo, F. H.. Industrial surface defect detection using Mask R-CNN with improved FPN. Manufacturing Automation, 42(12), 35-40+97.

[18] LWang Q, Wu B, Zhu P, et al. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks.2019 [2020-04-07].

[19] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[J]//IEEE Transactions on Pattern Analysis and Machine Intelligence,2020,42 (2):318-327.