Journal of Intelligent & Robotic Systems (2022) 104: 41
https://doi.org/10.1007/s10846-022-01580-w

REGULAR PAPER

Palletizing Robot Positioning Bolt Detection Based on Improved YOLO-V3

Ke Zhao[1] · Yaonan Wang[1] · Yi Zuo[2] · Chujin Zhang[1]

Received: 24 September 2021 / Accepted: 20 January 2022 / Published online: 24 February 2022

Abstract

To improve the detection accuracy and speed of palletizing robot positioning bolts in complex scenes, we proposed a positioning bolt (PB) detection method based on improved YOLO-V3. First, due to the actual detection requirement, we constructed the PB data set by using a series of data enhancement operations such as horizontal flip, ±30degree rotation, and random luminance enhancement or decrease. Then, an improved anchor box mechanism based on the k-means++ algorithm was designed to obtain a more accurate anchor box for the PB data. According to the feature of the PB data in the palletizing robot, such as the existence of dust and dirt on the surface, the feature extraction network was further enhanced by adding a Densenet-4 module. In this way, the low-level semantics and high-level abstract features can be extracted effectively to improve detection performance. Finally, a new bounding box regression loss function was elaborated to accelerate the neural network training. The experimental results demonstrated the effectiveness of the proposed improvement mechanisms. The comparable results also show that our method is superior to the original YOLO-V3, SSD, and Faster R-CNN for PB data, and has a detection AP of 86.7%, a recall rate of 97%, and a detection speed of 25.47 FPS, which can achieve high-efficiency and high-precision detection in complex industrial scenarios.

Keywords Palletizing robot · Positioning bolt · YOLO-V3 · Densenet-4

1 Introduction

Palletizing robots are important equipment in the logistics system of modern manufacturing enterprises. They are mainly used for the acquisition and handling of large quantities of workpieces, which provide an important guarantee for the flexible and efficient operation of industrial systems [1]. The working process of industrial robots is generally to grab materials from a fixed position and transport them to a specific location for installation and other processes. Therefore, the industrial robot has extremely high requirements for the accuracy of the material position. Once the material position deviates slightly, it affects the material grabbing of the palletizing robot and even leads to the deviation of the final material installation position, thereby affecting the product quality (Fig. 1).

In the beginning, most industrial robots acquired the material position through manual pre-teaching, but the types of materials on the assembly line are diverse. When the materials change, the robots need to be re-teaching by professionals, which largely affects the flexibility and productivity of the robots, and increases operating costs [2].

To solve these problems, domestic and foreign researchers have combined machine vision and industrial robot control technology for material positioning [3]. Wang et al. [4] proposed a material positioning method by fusing scale invariant feature transform and moment invariants. Chen et al. [5] realized the material positioning by extracting the Hu invariant matrix of the material contour, but this method is extremely sensitive to material contour changes caused by factors such as occlusion, reflection, and shadow. Huang et al. [6] determined the material location by matching the extracted contours with the object contour from the template image, but this method cannot adapt to the change of complex environment. M et al. [7] used the aggregation and representativeness of surface features to identify

✉ Ke Zhao
dict123@163.com

[1] National Engineering laboratory for robot visual perception & control technology, Hunan University, Changsha, 410082, China

[2] Hunan University of Finance and Economics, Changsha, 410082, China

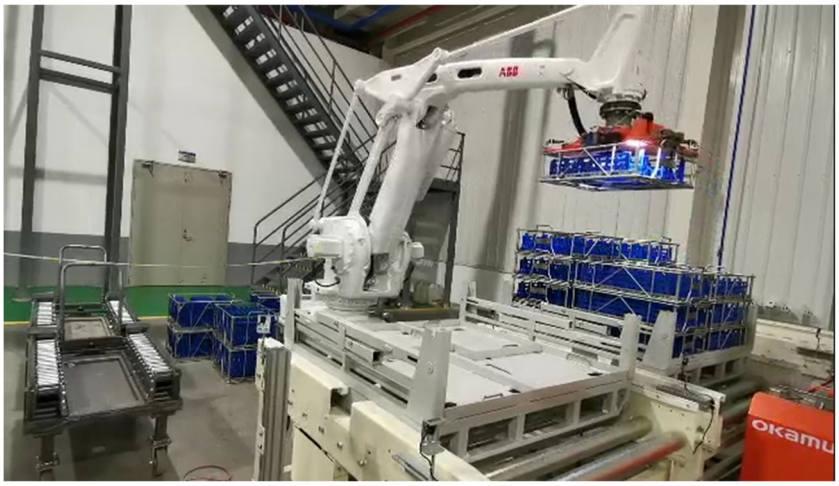

Fig. 1 Palletizing Process

materials. Choi et al. [8] took surface-to-surface, surface-to-boundary, boundary-to-boundary, and line-to-line as pair recognition features to identify material's location. To achieve the goal of the robot which has the capability of learning and grasping the predetermined workpieces actively on the assembly line, Yang et al. [9] proposed a multi-material detection algorithm based on the shape-SVM learning model. For incomplete information material detection, Fu et al. [10] proposed an algorithm of characteristic region minimum rectangle fitting to achieve the pose of assembly workpiece with incomplete information. Although the above solution achieves semi-automation of material positioning, it still requires human intervention and the accuracy needs to be improved.

In recent years, convolutional neural network (CNN) methods have achieved excellent results in many computer vision applications, such as object detection, natural language processing, image recognition, and automatic drive [11–14, 26] and simultaneous localization and mapping [33, 34]. Several workpiece detection methods based on CNN have also been proposed. Li et al. [15] combined a binocular eye-in-hand system and CNN for workpiece localization. Lin et al. [16] extracted the geometric features of the workpiece using CNN and k-means clustering, and then applied a particle swarm optimization algorithm to detect the degree of matching between the geometric model and the actual workpiece. Although these methods have been widely used in robot welding, trajectory tracking, and defect detection, there is still a lack of a high-precision material positioning method for palletizing.

Therefore, we proposed a Positioning Bolt (PB) detection algorithm based on improved YOLO-V3 [17], which can obtain the location of the PB and the material automatically with satisfactory accuracy. The specific contributions of this work are as follows:

1) According to the characteristics of the PB data, we designed a new anchor box mechanism based on k-means++ [18] to obtain the anchor box size suitable for PB.
2) Aiming at the problem that PB features are difficult to extract, we referred to the dense network structure in DenseNet [19], and designed densenet-4 module for feature extraction network to obtain richer semantic information of PB.
3) To accelerate the convergence of the network and improve the detection accuracy, a new bounding box loss function was proposed.

The remainder of this paper is organized as follows. In Section 2, we introduced the PB data set and the basic theory of YOLO-V3, In Section 3, we described the improved method of our approach in detail. Section 4 gave the experiments of our method on the PB data set and compared the performance with Faster R-CNN, SSD, and YOLO-V3. Finally, conclusions were presented in Section 5.

2 Related Work

2.1 Positioning Bolt Data Set

In this study, the PB image acquisition was conducted using a camera with 1920×1080 pixel resolution. We collected 1000 PB images under different lighting conditions, including 250 images from the front light, 250 images from the back light, 250 images from the side light, and 250 images from scattered light. In addition, considering that the camera view angle affects the detection performance, we collected 250 images from multiple view angles (directly above the PB, 45-degree angle to the left of the PB, and a 45-degree angle to the right of the PB) during the image

acquisition. These 1250 images were then expanded to 6000 images using data augmentation methods (flip, rotation, luminance enhancement). Several sample images in the obtained data set are shown in Fig. 2.

To better compare the performance of different algorithms, we used Labellmg to mark the position of PB in the image and converted it to Pascal VOC 2012 format for subsequent network training [20]. When constructing the data set, the length of the PB image is rescaled to 512 pixels and the width is adjusted accordingly to maintain the original aspect ratio. To ensure the correctness of the data set, we did not label the PB images with unclear imaging areas and we also did not label the images with PB occlusion areas greater than 85%.

2.2 YOLO-V3

PB detection is an object detection task within the field of computer vision. Recently, the rapid development of convolutional neural networks (CNN) and computer hardware afforded deep learning to become an essential asset for object detection [11], involving two-staged and one-staged methods. Two-staged algorithms are based on regional recommendation, i.e., in the original image the candidate region of the object to be detected is generated, and then on each candidate region, object identification and location are performed. In 2013, Girshick et al. [27] proposed the first region-recommendation-based object detection algorithm entitled R-CNN, which generated 2000 object candidate regions using selective search and then employed CNN to extract features from the candidate regions. On this basis, in [28], the authors proposed the Fast R-CNN variant, where a CNN is applied to the original image to obtain the feature map and links the candidate regions to the feature map to realize the sharing of candidate regions to the deep convolutional layer. This strategy solves the feature extraction requirement per candidate region, significantly reducing the computational burden. Subsequently, Ren et al. [24] developed the Faster R-CNN algorithm, involving a new candidate region generation network (RPN), solving the candidate box pre-generation problem and affording GPU acceleration.

One-staged algorithms rely on regression and do not require generating the target's candidate region, as these methods directly employ regression for object recognition and localization. The "one-blade flow" detection algorithms are represented by the YOLO and SSD families. In 2015, Redmon et al. [29] proposed YOLO, which directly solves the object detection problem as a regression task without generating candidate regions. Despite YOLO being very fast in detection, it is less robust and generalizable for small targets. Liu et al. [25] proposed SSD to address the above problems, which extracts feature maps of different scales and uses a priori frames of different scales and aspect ratios. Subsequent researchers improved YOLO and SSD and successively proposed a series of "one-stage" detection algorithms such as YOLO-V2, YOLO-V3, FSSD, RSSD, DSSD, and RetinaNet to further improve detection accuracy [30–32].

To balance the detection speed and accuracy of PB and apply it to industry, yolov3 is improved in this paper. YOLO-V3 regards object detection as a regression problem, and the network structure is shown in Fig. 3. Firstly, YOLO-V3 extracts multi-level image features by the Darknet-53 framework, which is composed of five residual modules, and each residual module is composed of one or more residual units. After obtaining the feature map, YOLO-V3 then selects three image feature layers to build a feature pyramid structure and uses the three feature layers output by the feature pyramid to predict the bounding box of the object and classify the object. The specific training process of YOLO-V3 is as follows.

Step 1: The image is scaled to a standard size of 416×416 and inputted into the network.

Step 2: Darknet-53 extracts image features and generates a 13×13 feature map on a small scale.

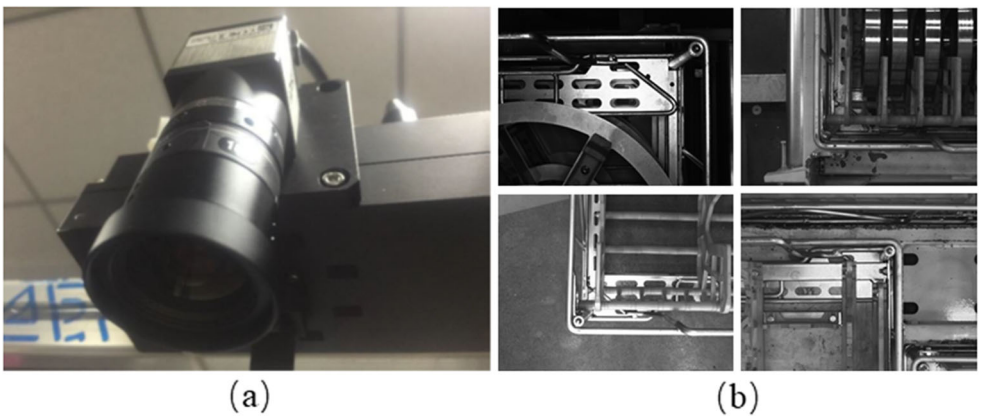

Fig. 2 (a) Palletizing Robot Camera (b) PB images

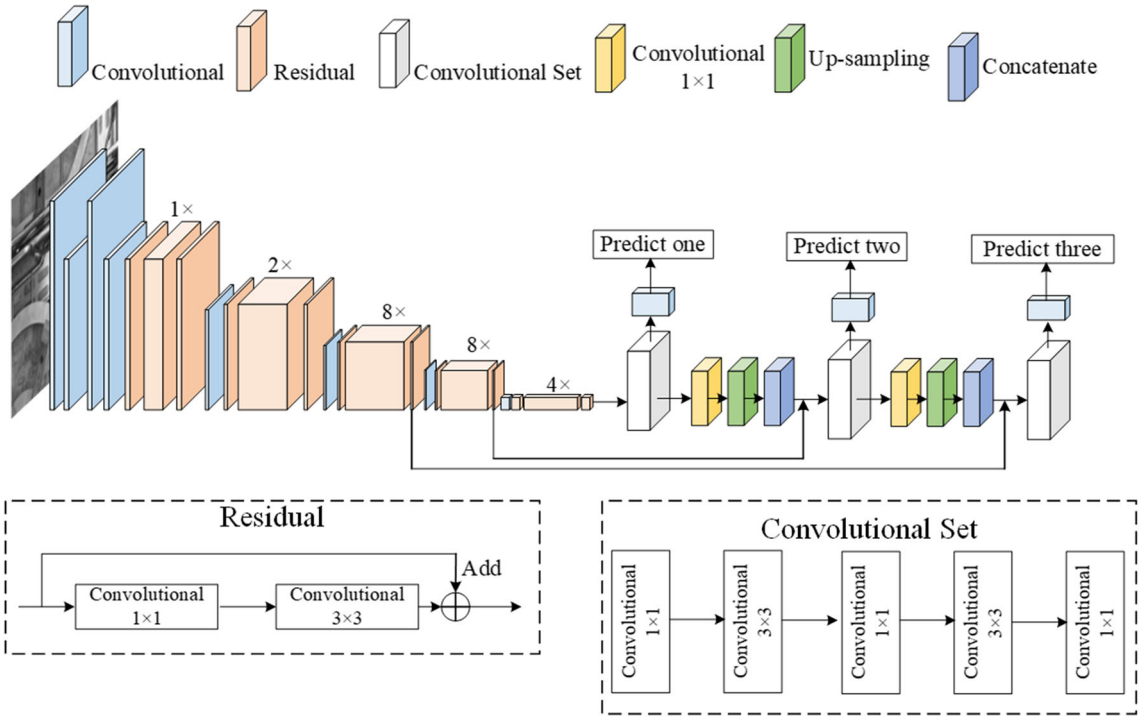

Fig. 3 The Network Structure of YOLO-V3

Step 3: The 13×13 small-scale feature map is first subjected to convolution set and 2 times up-sampling, and then connected with the 26*26 feature map to output the prediction result.

Step 4: the 26×26 feature map obtained by step3 is subjected to convolutional set and 2 times up-sampling, then connected to the 52×52 feature map and output the prediction result.

Step 5: The features from the three-scale prediction output are fused, and then the probability score is used as a threshold to filter out most anchor boxes with lower scores. Then use non-maximum suppression (NMS) for post-processing, leaving more accurate bounding boxes.

3 Proposed Method

Although YOLO-V3 performs well on public data sets such as voc2012 and COCO2017, it is not suitable for the PB images with blurred boundaries, greater noise, and lower contrast. To solve these problems, we improved YOLO-V3 in three aspects for the features of the PB dataset: 1) Used the k-means++ algorithm to obtain a more accurate PB anchor box size; 2) Added Densenet-4 to enhance the feature extraction network structure; 3) Used a new object bounding box regression loss function. The specific improvement measures are as follows.

3.1 New Anchor Mechanism

The original Yolo-V3 introduces the anchor frame mechanism and uses k-means clustering to obtain the size of the anchor frame. The anchor frame size obtained by K-means clustering effectively improves the detection performance of the target, but the anchor frame size in Yolo-V3 is obtained based on 80 targets such as people, bicycles, cars, and airplanes, which is not suitable for PB data evidently. Moreover, the clustering results of k-means are greatly affected by the selection of initial points. To solve this problem, this paper proposes a new anchor mechanism (NAM) that uses the K-means++ algorithm to cluster the positioning bolts to obtain the anchor frame size. The specific process is as follows:

1) Move the centers of all manually marked rectangular boxes(Ground Truth, GT) in the PB data set to the origin of the coordinate system.
2) Initialize a cluster center randomly, calculate the shortest distance D(X) between each sample and the currently known cluster center, and then calculate the probability of each sample being selected as the next cluster center; finally, follow the roulette method to select the next cluster center.
3) Repeat step 2) until k cluster centers are selected, that is, randomly select K rectangular boxes as cluster centers.

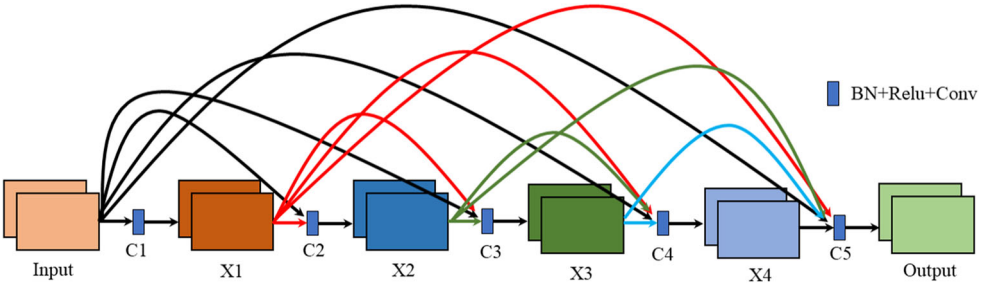

Fig. 4 The structure of DenseNet

4) Calculate the distance from each rectangular box to the K cluster centers according to formula (1), and classify each rectangular box to the nearest cluster center.

$$d(box,\ box_cluster) = 1 - IoU(box,\ cluster) \tag{1}$$

where $IoU(box,\ cluster)$ represents the intersection ratio of GT and cluster centers.

5) Recalculate the cluster centers of the k clusters after classification according to formula (2).

$$W' = \frac{1}{N}\sum w_i,\ H' = \frac{1}{N}\sum h_i \tag{2}$$

w_i represents the width of the i-th rectangle, h_i represents the height of the i-th rectangle, N is the number of rectangles in each cluster.

6) Repeat steps 4) and 5) until the cluster center change is less than the artificially set threshold and then stop the iteration.

Compared with k-means, k-means++ abandons the theory of randomly selecting k cluster centers, but randomly initializes a cluster center and selects subsequent cluster centers through the roulette method, so that the distance between the K cluster centers is far enough. Although K-means++ is time-consuming to calculate the clustering center, it can converge faster in the iterative process, which improves the network training speed. The 9 anchor frame sizes obtained by k-means++ in this paper are (20, 27), (22, 29), (28, 38), (41, 49), (45, 56), (52, 51), (52, 62), (59, 57), (61, 69).

3.2 Improved Feature Extraction Network

The detection object in this paper is the PB. Compared with the object in public data sets such as voc2012 and coco2017, the PB not only has a smaller size, but also has a dirt and ash layer on the surface. Its' features are difficult to extract, which leads to a decrease in the detection accuracy of the PB. To address this problem, we are inspired by DenseNet to improve Darknet-53. Darknet-53 is basically composed of 1×1 or 3×3 convolutional kernels, while several ResNet [21] are used, but the ResNet use superposition to handle constant mappings and nonlinear outputs, which disrupts the information flow in the network to some extent. Unlike ResNet, which adds the values of the subsequent layers by constructing an identity map, DenseNet connects all the layers for channel merging to achieve feature reuse. Compared with ResNet, the backpropagation of the gradient is enhanced, which can make better use of feature information and improve the transmittance of the information between layers.

In Fig. 4, X1, X2, X3, and X4 represent the feature maps, while C1, C2, C3, C4 and C5 refers to the nonlinear transformations (Batch Normalization+Relu+Convolutional (BN+Relu+Conv)) [22]. DenseNet connects each layer to other layers in feedforward mode, thus layer l receives all the feature maps of the preceding layers $x_0, x_1, x_2...x_{l-1}$ as input.

$$x_l = C_l[x_0, x_1, x_2...x_{l-1}] \tag{3}$$

Therefore, we designed the network structure named densenet-4 by referring to the idea of Densenet, which consists of 4 DCBR modules. The convolution, Batch Normalization, and ReLU make up the CBR module, while two CBR modules are cascaded into a Double-CBL (DCBR) module. We used the DCBL module as transport layer Ci : Conv (1 × 1 × 32)-BN-ReLU-Conv (3 × 3 × 64)-BN-ReLU and Conv (1 × 1 × 64)-BN- ReLU- Conv (3 × 3 × 128)-BN-ReLU. To balance the detection speed and accuracy, we kept the residual modules with output sizes of 208×208 and 104×104 in the Darknet-53 framework, and replace the residual modules with outputs of 52×52, 26×26, and 13×13 with densenet-4 modules. The feature extraction network of our method is shown in Fig. 5.

3.3 Bounding Box Regression Loss Function

The loss function of Yolo-V3 consists of the object bounding box regression loss $L_{loc}(l, g)$, the object confidence loss $L_{conf}(o, c)$ and the object classification loss $L_{cla}(O, C)$, where λ_1, λ_2 and λ_3 are the balanced weight coefficients.

$$L(O, o, C, c, l, g) = \lambda_1 L_{conf}(o, c) + \lambda_2 L_{cla}(O, C) + \lambda_3 L_{loc}(l, g) \tag{4}$$

	Type	Filters	Size	Output
	Convolutional	32	3×3	416×416
	Convolutional	64	3×3/2	208×208
1×	Convolutional Convolutional Residual	32 64	1×1 1×1	208×208
	Convolutional	128	3×3/2	104×104
2×	Convolutional Convolutional Residual	64 128	1×1 3×3	104×104
	Convolutional	256	3×3/2	52×52
8×	Convolutional Convolutional Densenet-4	128 256	1×1 3×3	52×52
	Convolutional	256	3×3/2	26×26
8×	Convolutional Convolutional Densenet-4	256 512	1×1 3×3	26×26
	Convolutional	1024	3×3/2	13×13
4×	Convolutional Convolutional Densenet-4	512 1024	1×1 3×3	13×13

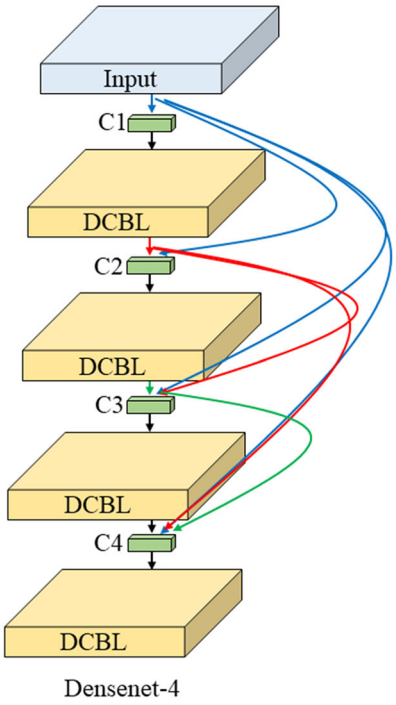

Fig. 5 The feature extraction network structure of our algorithm

The bounding box regression loss function is shown in Eq. 5, where(x, y) represents the center coordinates of the bounding box, w and h are the width and height of the rectangular box, respectively, $\hat{l_i^m}$ denotes the coordinate offset of the m^{th} predicted bounding box, and $\hat{g_i^m}$ denotes the coordinate offset of the m^{th} Ground Truth (GT).

$$L_{loc}(l, g) = \sum_{i \in \text{pos}} \sum_{m \in \{x,y,w,h\}} \left(\hat{l_i^m} - \hat{g_i^m}\right)^2 \tag{5}$$

From Eq. 5, we can see that the mean square error (MSE) is used to calculate the regression loss of the bounding box in Yolo-V3. However, MSE is sensitive to the change of object scale, and the large size of the bounding box would generate more loss values, which brings difficulties to optimize the small size bounding box, resulting in poor detection of small size objects.

To address this problem, we adopt IoU to calculate the bounding box regression loss [23]. IoU is usually used to measure the relative size of the overlap area between the target prediction box and the real box, and it has scale invariance, non-negativity, and symmetry. However, there remains two problems when directly using IoU as the bounding box regression loss function:1) When the value of IoU is 0, the value of the regression loss function is also 0. At this time, the network cannot return the gradient and cannot update the parameters; 2) IoU cannot accurately reflect the overlap between the object prediction box and the GT. As shown in the Fig. 6, the IoU value is the same, but the overlap of the object prediction box and the GT from left to right is different.

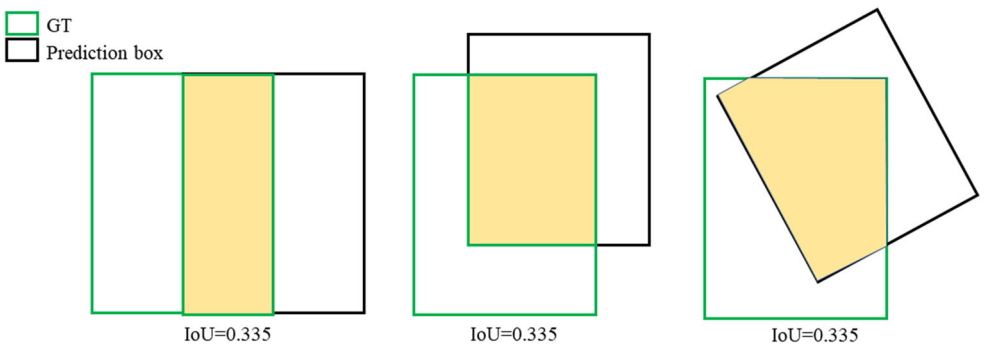

Fig. 6 Different overlaps of the same IoU value

Therefore, we proposed a new bounding box regression loss function, which was defined as follows.

$$L_{loc} = 1.01 - IoU(B^P, B^G) + \frac{w^d \times h^d}{w^c \times h^c} \tag{6}$$

where B^P is the object prediction box, B^G is the object GT, w^d and h^d are the width and height of the rectangular box B^d enclosed by the center point and the center point of B^G and B^p, w^c and h^c are the width and height of the minimum closed box B^c for B^G and B^P, respectively.

The specific calculation procedure is shown in Table 1. L_{loc} added a penalty term $\frac{w^d \times h^d}{w^c \times h^c}$ to the IoU, since the area of B^c is always larger than the area of B^d, so: $0 \leq \frac{w^d \times h^d}{w^c \times h^c} \leq 1$. When the object prediction box overlaps with the GT: $\frac{w^d \times h^d}{w^c \times h^c} = 0$, $L_{loc} = 0.01$; when the object prediction box does not intersect with the GT: $\frac{w^d \times h^d}{w^c \times h^c} = 1$, $L_{loc} = 2.01$.

Table 1 Imporved Bounding Box Regression Loss Function

Algorithm 1 L_{loc} as bounding box loss

Input: Object prediction box coordinates: $B^p = \left(x_1^p, y_1^p, x_2^p, y_2^p\right)$,
Object GT coordinates: $B^g = \left(x_1^g, y_1^g, x_2^g, y_2^g\right)$
Output: L_{loc}
1: Calculate the areas of B^p and B^g : S^p and S^g

$$S^p = |\left(x_2^p - x_1^p\right) * \left(y_2^p - y_1^g\right)| \tag{7}$$

$$S^g = |\left(x_2^g - x_1^g\right) * \left(y_2^g - y_1^g\right)| \tag{8}$$

2: Calculate the area of the overlapping area of B^p and B^g : S^i

$$S^i = \begin{cases} |\left(x_2^i - x_1^i\right) * \left(y_2^i - y_1^i\right)| & x_2^i > x_1^i, y_2^i > y_1^i \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where the value of x_1^i, x_2^i, y_1^i and y_2^i are::

$$\begin{aligned} x_1^i &= \max\left(x_1^p, x_1^g\right), x_2^i = \min\left(x_2^p, x_2^g\right) \\ y_1^i &= \max\left(y_1^p, y_1^g\right), y_2^i = \min\left(y_2^p, y_2^g\right) \end{aligned} \tag{10}$$

3: Calculate $IoU(B^P, B^G)$:

$$IoU(B^P, B^G) = \frac{S^i}{S^p + S^g - S^i} \tag{11}$$

4: Find the coordinates of B^c of the smallest closed box of B^p and B^g:

$$\begin{aligned} x_1^c &= \min\left(x_1^p, x_1^g\right), x_2^c = \max\left(x_2^p, x_2^g\right) \\ y_1^c &= \min\left(y_1^p, y_1^g\right), y_2^c = \max\left(y_2^p, y_2^g\right) \end{aligned} \tag{12}$$

5: Calculate the width and height of B^c:

$$w^c = x_2^c - x_1^c, \; h^c = h_2^c - h_1^c \tag{13}$$

6: Calculate the coordinates of the center points of B^p and B^g : b^p, b^g

$$b^p = \left(\frac{x_1^p + x_2^p}{2}, \frac{y_1^p + y_2^p}{2}\right), \; b^g = \left(\frac{x_1^g + x_2^g}{2}, \frac{y_1^g + y_2^g}{2}\right) \tag{14}$$

7: Calculate the coordinates of B^d:

$$x_1^d = \min\left(\frac{x_1^p + x_2^p}{2}, \frac{x_1^g + x_2^g}{2}\right), \; x_2^d = \max\left(\frac{x_1^p + x_2^p}{2}, \frac{x_1^g + x_2^g}{2}\right) \tag{15}$$

$$y_1^d = \min\left(\frac{y_1^p + y_2^p}{2}, \frac{y_1^g + y_2^g}{2}\right), \; y_2^d = \max\left(\frac{y_1^p + y_2^p}{2}, \frac{y_1^g + y_2^g}{2}\right) \tag{16}$$

8: Calculate the width w^d, height h^d of B^d:

$$w^d = x_2^d - x_1^d, \; h^d = y_2^d - y_1^d \tag{17}$$

9: Calculate L_{loc}:

$$L_{loc} = 1.01 - IoU(B^P, B^G) + \frac{w^d \times h^d}{w^c \times h^c} \tag{18}$$

In summary, when IoU = 0, L_{loc} can still reflect the relative distance between the object prediction box and the GT, and provide the moving direction for the regression of the object prediction box.

4 Experiments and Discussions

4.1 Experimental Setup

The dataset used in this experiment is derived according to Section 2.1, and the dataset is divided into three parts: training set (4000 images), test set (500 images) and the validation set (500 images).

a) Experimental environment: The experimental environment for model training and verification in this paper is shown in the Table 2.

b) Training parameters: In the training process of the PB detection network, the maximum number of training steps is 100 Epoch; the learning rate is set to be 0.0001 and divided by 10 at 20 epoch, 40 epoch, 60 epoch and 80 epoch; the training/test image size is 512×512 and the batch size is 8; the MBGD is applied to minimize the loss function.

c) Evaluation indicators: We evaluate the method in terms of average precision (AP), and processing time costs. True positives (TP), false negatives (FN) and false positives (FP) are firstly calculated to generate evaluation metrics including recall and precision. Recall is used to measure the completeness of the test results, while precision is used to indicate accuracy. The indexes are defined as follows.

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

AP is then obtained by calculating the area under the Precision-Recall (PR) curve according to Eq. 21. Frames per second (FPS) is usually used to measure the cost of

Table 2 Experimental environment

Platform	Type
CPU	Intel Core i9-9900KF
GPU	Nvidia GeForce RTX 3090
Memory	24GB
Operating System	Ubuntu 18.04
Deep Learning Framework	Pytorch1.4
Programming Language	Python3.6

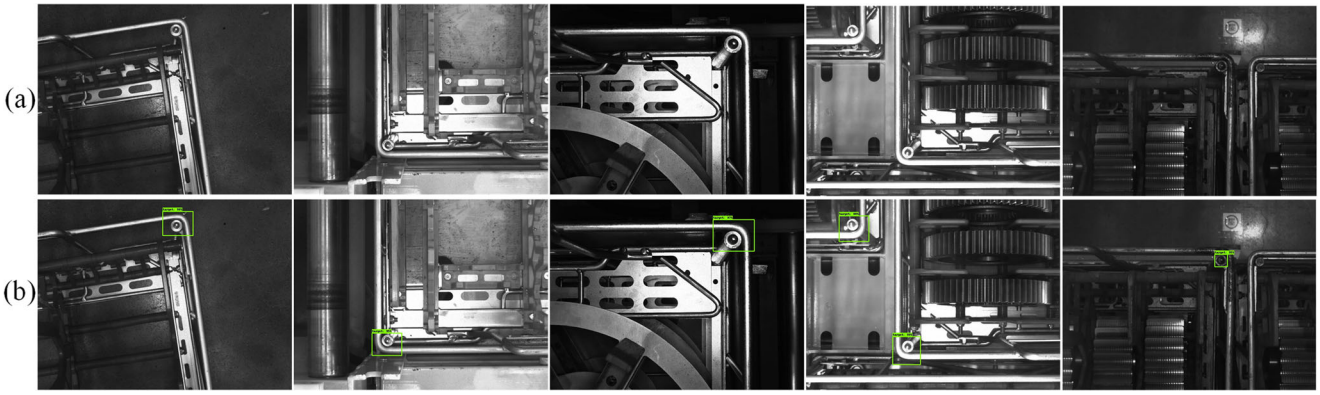

Fig. 7 Fig. 7(a) Origin images (b) PB Detection results

processing time and is defined as the number of images that the training model detected in 1 second.

$$AP = \int_0^1 PRdr \tag{21}$$

4.2 Experimental Results

The partial detection results of our method on the PB data set are shown in Fig. 7. Under different illumination, different angles and different distributes, our method can accurately detect PB in the image, which proves that our method has great detection performance for the PB.

To verify the accuracy and real-time performance of our method, we compared our approach with Faster R-CNN [24], SSD [25], and YOLO-V3. The training epoch is an important factor that affects the performance of deep networks. Sufficient training epoch ensures the convergence of the entire training process and enables the module to achieve the best performance under certain parameter configuration. Figures 8 and 9 showed the Precision and Loss of the above method in the training process. It can be seen that convergence is achieved for each method. Specifically, our method converges after about the 20th epoch, while YOLO-V3, SSD and Faster R-CNN have more severe oscillations during training, and they need about 45 epochs to reach convergence. The precision and loss values of our method after 100 epochs are about 0.97 and 0.6, while yolo-v3 and Faster R-CNN both have lower precision than ours, about 0.93 and 0.82, respectively, and SSD has the lowest precision (about 0.78) and highest loss value (about 1.7)

Table 3 shows that our method outperforms SSD and Faster R-CNN in AP and FPS indicators. Compared with

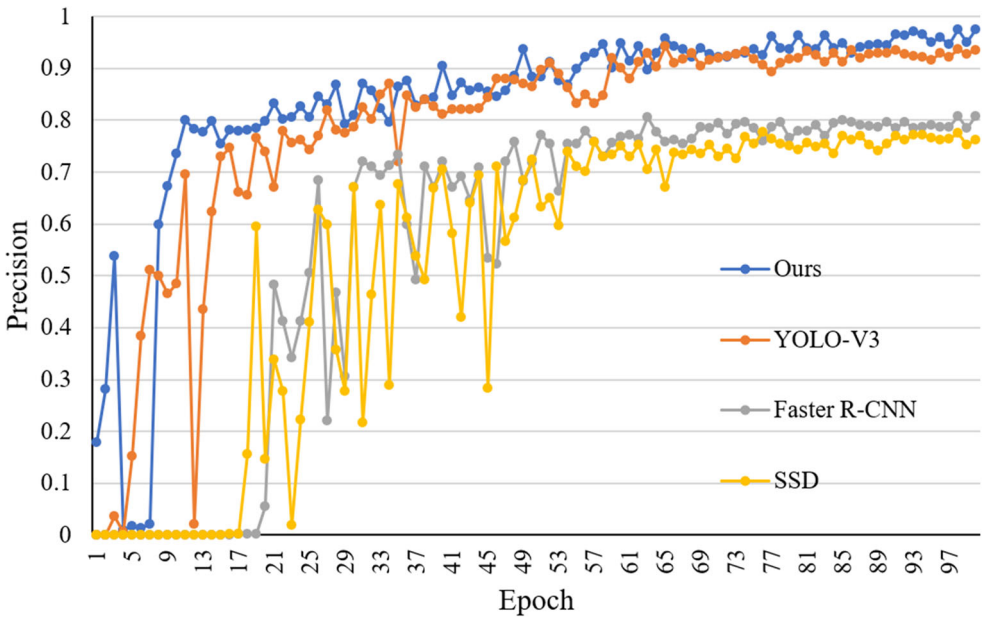

Fig. 8 Variation of Precision with respect to training epochs

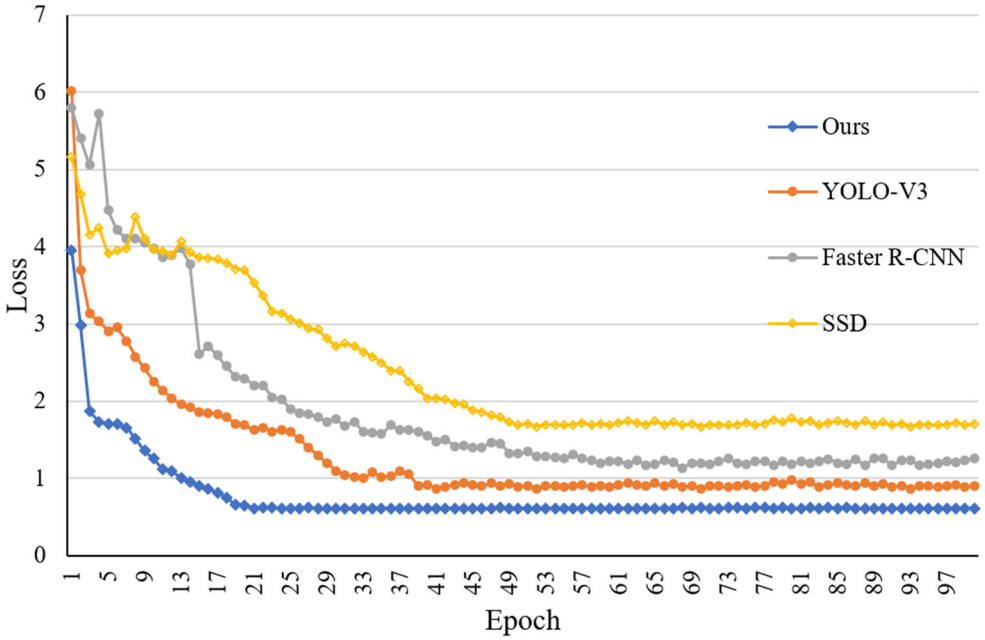

Fig. 9 Variation of Loss with respect to training epochs

YOLO-V3, the detection speed of our method is not significantly reduced, but there is a significant improvement in the detection AP of PB. The experimental results show that our improved YOLO-V3 can effectively detect PBs in real industrial scenes under real-time detection.

4.3 Ablation Experiments

In this section, the effectiveness of each improved module is investigated. The results of the ablation experiments are shown in Table 4. The second method adds the NAM module, resulting obvious improvement in AP from 80.8 to 81.2 and detection speed from 27.13 FPS to 27.58 FPS. The feature extraction network of the third method introduces the DenseNet-4 module based on DarkNet-53, and the result in AP improves from 80.8 to 83.6. The bounding box loss function L is applied in the fourth method, whose AP improves from 80.8 to 83.7 compared to the original YOLO-V3. The experimental results show that the proposed three modules can improve the detection performance of PB with essentially the same detection speed.

4.4 Expansion Experiment

To more intuitively verify the effectiveness of our method, we selected several representative images and compare the detection results with Faster R-CNN, SSD and YOLO-V3. The comparison of detect results are shown in Fig. 10.

We can see that for images with darker lighting conditions (such as Sample 2, Sample 6), Faster R-CNN and SSD have missed detection, and PB cannot be detected. For images with interfering objects (such as Sample1, Sample3), Faster R-CNN and SSD have false detection. Although YOLO-V3 performs better than Faster R-CNN and SSD, there are still false detection (Sample1, Sample2). In contrast, our method accurately detected all PBs.

A series of comparative experiments show that the detection performance of PB in complex backgrounds can

Table 3 Experimental comparisons of different methods

Method	Backbone	Number	AP(IoU=0.5)	FPS
Faster R-CNN	VGG-16	500	78.4	14.61
SSD	VGG-16	500	76.2	17.27
YOLO-V3	DarkNet-53	500	80.8	27.13
Ours	Fig. 5	500	86.7	25.47

Table 4 Comparison results of ablation experiments

Method	Backbone	Number	AP(IoU=0.5)	FPS
YOLO-V3	Darknet-53	500	80.8	27.13
YOLO-V3+NAM	Darknet-53	500	81.2	27.58
YOLO-V3 +DenseNet-4	/	500	83.6	26.14
YOLO-V3+L_{loc}	Darknet-53	500	83.7	26.77

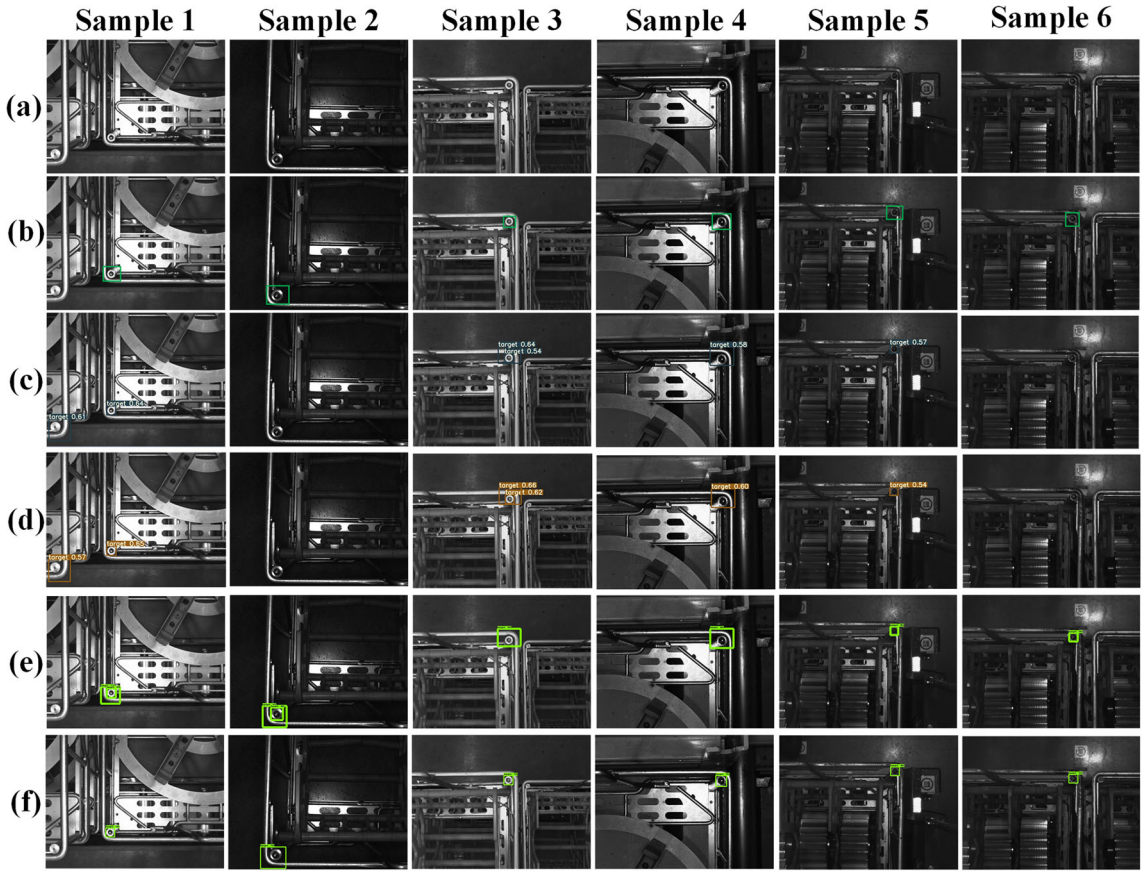

Fig. 10 Expansion Experiment (a) Origin images (b) Ground Truth (c) Faster R-CNN (d) SSD (e) YOLO-V3 (f) Ours

be improved by improving the anchor box size, feature extraction network and loss function.

5 Conclusion

For the small size of the PB in the palletizing robot and the existence of dust and dirt on the surface, we have proposed an intelligent detection method for the PB based on YOLO-V3.

This method first re-clusters and optimizes the anchor box size of the YOLO-V3 model to obtain the anchor box size suitable for PB detection, while introducing the Densenet-4 module in the feature extraction stage, which is a structure capable of obtaining the low-level semantics and high-level abstract features of the PB, and finally a bounding box regression loss function is designed.

Our method has realized high-efficiency and high-precision PBs detection, which provided a guarantee for the accurate identification of subsequent material positions. The experimental results showed that the algorithm in this paper can achieve the AP of 86.7%, the recall rate of 97%, and a detection speed of 25.47 FPS, which is higher than Faster R-CNN and YOLO-V3, and can meet the high-efficiency and accurate detection of PB in real industrial complex environments.

Acknowledgements The authors would like to acknowledge the support of the National Natural Science Foundation of China - Key Project 61733004, 62027810, 62076091 and 62133005.

Author Contributions The overall study supervised by Yaonan Wang; Methodology, hardware, software, and preparing the original draft by Ke Zhao; Review and editing by Qing Zhu and Yi Zuo; The results were analyzed and validated by Chujin Zhang. All authors have read and agreed to the published version of the manuscript.

Code Availability Code generated or used during the study is available from the corresponding author by request.

Declarations

Conflict of Interests All the authors of this paper have no conflicts of interest, financial or otherwise.

References

1. Moura, F.M., Silva, M.F.: Application for automatic programming of palletizing robots. In: 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 48–53. IEEE (2018)

2. de Souza, J.P.C., Castro, A.L., Rocha, L.F., et al.: Adaptpack studio translator: translating offline programming to real palletizing robots Industrial Robot: The International Journal of Robotics Research and Application (2020)
3. Li, C., Ma, Y., Wang, S., et al.: Novel industrial robot sorting technology based on machine vision. In: 2017 9th International Conference on Modelling, Identification and Control (ICMIC), pp. 902–907. IEEE (2017)
4. Wang, J., Zhang, X., Dou, H., et al.: Study on the target recognition and location technology of industrial sorting robot based on machine vision. J. Robot. Netw. Artif. Life **1**(2), 108–110 (2014)
5. Chen, Z.N., Zhang, X., Peng, Z.R., et al.: Workpiece location and recognition based on machine vision. Electron. Sci. Technol. **29**(4), 99–103 (2016)
6. Huang, C., Chen, D., Tang, X.: Implementation of workpiece recognition and location based on opencv. In: 2015 8th International Symposium on Computational Intelligence and Design (ISCID), vol. 2, pp. 228–232. IEEE (2015)
7. Jinqiu, M., Tongshuai, Z., Zhiyu, Z.: An Approach for Picking T-shape workpiece based on monocular vision. In: 2018 3rd International Conference on Information Systems Engineering (ICISE), pp. 1–5. IEEE (2018)
8. Choi, C., Taguchi, Y., Tuzel, O., et al.: Voting-based pose estimation for robotic assembly using a 3D sensor. In: 2012 IEEE International Conference on Robotics and Automation, pp. 1724–1731. IEEE (2012)
9. Yang, L., Chong, M., Bai, C., et al.: A multi-workpieces recognition algorithm based on shape-SVM learning model. J. Phys. Conf. Series. IOP Publishing **1087**(2), 022025 (2018)
10. Fu, T., Li, F., Zheng, Y., et al.: Dynamically grasping with incomplete information workpiece based on machine vision. In: 2019 IEEE International Conference on Unmanned Systems (ICUS), pp. 502–507. IEEE (2019)
11. Zhao, Z.Q., Zheng, P., Xu, S., et al.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. **30**(11), 3212–3232 (2019)
12. Wang, X., Liu, M., Raychaudhuri, D.S., et al.: Learning person Re-Identification models from videos with weak supervision. IEEE Trans. Image Process. **30**, 3017–3028 (2021)
13. Hu, H., Zhang, Z., Xie, Z., et al.: Local relation networks for image recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3464–3473 (2019)
14. Jiang, W., Liu, M., Peng, Y., et al.: HDCB-Net: A neural network with the hybrid dilated convolution for pixel-level crack detection on concrete bridges. IEEE Trans. Indust. Inform. **17**(8), 5485–5494 (2020)
15. Li, C.H.G., Chang, Y.M.: Automated visual positioning and precision placement of a workpiece using deep learning. Int. J. Adv. Manufact. Technol. **104**(9), 4527–4538 (2019)
16. Lin, X., Wang, X., Li, L.: Intelligent detection of edge inconsistency for mechanical workpiece by machine vision with deep learning and variable geometry model. Appl. Intell. **50**(7), 2105–2119 (2020)
17. Redmon, J., Farhadi A.: Yolov3: An incremental improvement. arXiv:1804.02767 (2018)
18. Kapoor, A., Singhal, A.: A comparative study of K-Means, K-Means++ and Fuzzy C-Means clustering algorithms. In: 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), pp. 1–6. IEEE (2017)
19. Huang, G., Liu, Z., Van Der Maaten, L., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
20. Zhou, P., Ni, B., Geng, C., et al.: Scale-transferrable object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 528–537 (2018)
21. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
22. Agarap, A.F.: Deep learning using rectified linear units (relu). arXiv:1803.08375 (2018)
23. Rezatofighi, H., Tsoi, N., Gwak, J.Y., et al.: Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658–666 (2019)
24. Ren, S., He, K., Girshick, R., et al.: Faster r-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Machine Intell. **39**(6), 1137–1149 (2016)
25. Liu, W., Anguelov, D., Erhan, D., et al.: Ssd: Single shot multibox detector. European conference on computer vision, pp. 21–37. Springer, Cham (2016)
26. Tang, Y., Li, B., Liu, M., et al.: Autopedestrian: an automatic data augmentation and loss function search scheme for pedestrian detection. IEEE Transactions on Image Processing (2021)
27. Girshick, R., Donahue, J., Darrell, T., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
28. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
29. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
30. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
31. Redmon, J., Farhadi A.: Yolov3: An incremental improvement. arXiv:1804.02767 (2018)
32. Lin, T.Y., Goyal, P., Girshick, R., et al.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
33. Wang, K., Ma, S., Chen, J., et al.: Approaches challenges and applications for deep visual odometry toward to complicated and emerging areas. IEEE Transactions on Cognitive and Developmental Systems (2020)
34. Wang, K., Ma, S., Ren, F., et al.: SBAS: Salient bundle adjustment for visual SLAM. IEEE Trans. Instrum. Meas. **70**, 1–9 (2021)

Ke Zhao received the M.S. degree in electrical engineering from Hunan University, Changsha, China, in 2011. He is currently pursuing the Ph.D. degree in control science and engineering from Hunan University. His current research interests include machine vision and deep learning.

Yaonan Wang received the Ph.D. degree in electrical engineering from Hunan University, Changsha, China, in 1994. Since 1995, he has been a Professor with the College of Electrical and Information Engineering, Hunan University. From 1994 to 1995, he was a Post-Doctoral Research Fellow with the Normal University of Defense Technology, Changsha. From 1998 to 2000, he was supported as a Senior Humboldt Fellow by the Federal Republic of Germany at the University of Bremen, Bremen, Germany. From 2001 to 2004, he was a Visiting Professor at the University of Bremen. He is a member of the Chinese Academy of Engineering. His research interests include robotics and image processing.

Yi Zuo received the Ph.D. in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2016. He is currently a teacher with Hunan University of Finance and Economics. His current research interests include recommender systems, neural networks, and deep learning.

Chujin Zhang received the M.S. degree in electrical engineering from Hunan University, Changsha, China, in 2015. His current research interests include machine vision and deep learning.