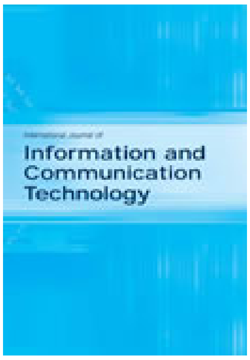


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Improved SSD-based visual sorting control for industrial robots

Wenke Yang, Guiyin Ran, Chao Yang

Article History:

Received:	27 August 2024
Last revised:	13 September 2024
Accepted:	13 September 2024
Published online:	17 October 2024

Improved SSD-based visual sorting control for industrial robots

Wenke Yang*, Guiyin Ran and Chao Yang

Chongqing Institute of Engineering,
Chongqing 400056, China
Email: m13608315776_1@163.com
Email: 2926482462@qq.com
Email: 1807244331@qq.com
*Corresponding author

Abstract: Traditional vision sorting methods for industrial robots often rely on rule-based image processing algorithms or simple deep learning models, which limit the system's recognition accuracy and real-time performance in complex environments. With the development of industrial automation and smart manufacturing, there is an increasing demand for vision sorting systems that can handle complex backgrounds and diverse targets. To address these issues, this work proposes an improved SSD-based vision sorting control system for industrial robots. Firstly, the VGG-16 backbone network of the conventional SSD is replaced with a GhostNet network to significantly enhance instant response capability of target recognition. Second, the SENet is brought to improve the accuracy of the target detection network. In addition, GhostNet is improved to enhance the flexibility of feature extraction by introducing dynamic convolution technique. The SENet module is optimised with a more efficient activation function to further reduce the computational complexity. Finally, the model's perceptiveness in identifying lesser targets is enhanced by a multi-scale feature fusion method. The experimental results show that compared with the original SSD model, the improved SSD model improves the mAP of target detection by 7.4% and the FPS by 28.4%.

Keywords: industrial robots; vision sorting; single shot multibox detector; SSD; GhostNet; SENet.

Reference to this paper should be made as follows: Yang, W., Ran, G. and Yang, C. (2024) 'Improved SSD-based visual sorting control for industrial robots', *Int. J. Information and Communication Technology*, Vol. 25, No. 8, pp.64–80.

Biographical notes: Wenke Yang received his Master's degree from Chongqing University of Technology in China in June 2020. Currently, He is a Lecturer at Chongqing Institute of Engineering. His research interests are robotics and machine vision.

Guiyin Ran received his Bachelor's degree from China Chongqing Institute of Engineering in June 2024. He is currently an Electrical Engineer in Chongqing Longwei Electromechanical Technology Co., Ltd.

Chao Yang received his Bachelor's degree from Chongqing Institute of Engineering in China in June 2024. He is currently an Electrical Engineer in Shenzhen Buke Electric Co., Ltd.

1 Introduction

Industrial robot vision sorting, as an important part of intelligent manufacturing (Lin et al., 2019; Shaikat et al., 2020), has a wide range of application prospects and important research value. With the continuous development of industrial automation, the traditional sorting method can no longer meet the needs of modern production lines for efficiency, precision and flexibility. Through the vision system, industrial robots can achieve automatic identification (Abbood et al., 2020), localisation and sorting of target objects, which can significantly improve the production efficiency and reduce the labour cost (Wu et al., 2019), and at the same time reduce the error rate. Therefore, the study of industrial robot vision sorting technology can not only promote the intelligent transformation of the manufacturing industry, but also produce significant economic and social benefits in terms of improving product quality and production efficiency.

Conventional robotic vision sorting methods mainly rely on rule-based image processing techniques, which usually pre-process and analyse images for target object detection and classification through specific algorithms such as edge detection, shape matching and colour segmentation (Kiyokawa et al., 2022). Vayda and Kak (1991) proposed a sorting method using shape matching sorting algorithm, which can achieve more accurate object recognition in simple environments. However, the recognition accuracy of this method decreases significantly in the case of complex backgrounds and diverse target objects, showing obvious limitations that make it difficult to meet the actual production requirements. Ciora and Simion (2014) used colour segmentation and template matching techniques for sorting industrial parts. This method works better in environments with stable colours and shapes, but the accuracy and efficiency are significantly reduced when dealing with targets with high colour similarity and complex shapes, which are susceptible to changes in ambient lighting. Charmette et al. (2016) used a feature-point-based image matching method for sorting, but this method is sensitive to rotations and scale changes of the target object, and is sensitive to changes in occlusion and complex backgrounds, the recognition performance decreases significantly, and it is difficult to cope with the variability of real industrial environments.

Currently, with the development of deep learning technology, visual sorting methods based on deep learning have become the mainstream of research. By training convolutional neural network (CNN) models, these methods are able to automatically learn and extract features in images, thus achieving efficient recognition and classification of target objects. Wang et al. (2018) proposed the you only look once (YOLO) model, which significantly improves detection speed and accuracy by transforming the target detection problem into a regression problem. It was found that although YOLO performs well in most cases, it is ineffective in detection and prone to miss detection when dealing with small and multi-scale targets. Wang et al. (2019) proposed a CNN model for target detection using multi-scale feature maps to achieve efficient real-time detection. It was shown that CNN is susceptible to noise interference in complex backgrounds and the detection accuracy is affected. Liu et al. (2024) proposed the addition of squeeze-and-excitation networks (SENet) module to the Yolov4 model, which significantly improves the feature representation capability of the target detection model by introducing the attention mechanism.

To cope with the above problems, this paper proposes an industrial robot vision sorting method based on an improved single shot multibox detector (SSD) model to enhance the recognition accuracy and real-time performance of the sorting system. Firstly, the VGG-16 backbone network in the traditional SSD algorithm is replaced with a GhostNet network to significantly increase the detection rate and enhance the real-time target recognition. Second, the SENet is used to improve the accuracy of target detection network and ensure detection accuracy. In addition, the shortcomings of GhostNet and SENet are respectively improved to further optimise the detection performance.

The main innovations and contributions of this work include:

1. Detection rate enhancement: to address the limitations of traditional SSD algorithms in terms of detection rate, this paper replaces the VGG-16 backbone network with a GhostNet network (Paoletti et al., 2021; Li et al., 2022). GhostNet effectively improves the computational efficiency through the reduction of redundant features, which results in a substantial enhancement in the rate of detection, thus enhancing the immediate response capability of the sorting system. This improvement is particularly significant in industrial environments where high frequency real-time detection is required.

2. Improvement of detection accuracy: in order to cope with the challenges of complex background and small target detection, this paper uses the SENet. SENet enhances the accuracy of feature extraction and reduces the interference of background noise by explicitly modelling the dependencies between feature channels, which enhances the precision of target detection. This improvement effectively enhances the recognition capability of the sorting system in complex environments.

3. Optimisation of the network structure: this paper improves on the basis of GhostNet by introducing the dynamic convolution technique, which adaptively adjusts the convolution kernel according to the input features to improve the flexibility and effectiveness of feature extraction. Meanwhile, the SENet module is optimised with a more efficient activation function, which further reduces the computational complexity and improves the model performance.

4. Multi-scale feature fusion: we introduce a multi-scale feature integration technique that bolsters the model's capacity for identifying minute targets through the adept amalgamation of feature maps across various scales. This approach not only elevates the precision of detection but also maintains the robustness of the sorting mechanism amidst a variety of target entities.

2 Relevant technologies

2.1 SSD network

SSD is a deep learning model for target detection (Yan et al., 2022), which is widely used in real-time detection tasks due to its speed and accuracy. The central principle behind SSD is the recasting of the detection of targets as a regression-based approach and directly predicts the class and bounding box location of the target by using a CNN.

The network structure of SSD mainly consists of a base feature extraction network and multi-scale feature layers. The base feature extraction network usually uses a pre-trained convolutional neural network, e.g., VGG-16. On this basis, SSD adds multiple convolutional layers to obtain feature maps at different scales for detecting targets of different sizes. Conventional SSDs use VGG-16 as the base network, removing the fully connected layers and retaining the convolutional layer portion to extract the features underlying the image. After the base network, the SSD adds multiple convolutional layers at different levels, which output feature maps with different spatial resolutions to facilitate detection of targets of different sizes. The specific convolutional layers include Conv4_3, Conv7, Conv8_2, Conv9_2, Conv10_2 and Conv11_2.

At each feature layer, SSD uses two convolutional layers (Ma et al., 2020), one for predicting categories (classification convolutional layer) and the other for predicting bounding box locations (localisation convolutional layer). At every point within the feature maps, the outputs from these convolutional layers produce several prediction frames, each with an associated category and bounding box adjustment. At each feature map location, the SSD defines multiple default boxes that have different aspect ratios and scales. By matching the real frames with the default frames during training, SSD can predict the category and bounding box offset of an object.

The loss function of SSD consists of two parts, classification loss and localisation loss, as shown below:

$$L(x,c,l,g)=\frac{1}{N}\left(L_{conf}(x,c)+\alpha L_{loc}(x,l,g)\right) \tag{1}$$

where L_{conf} is the classification loss, L_{loc} is the localisation loss, N is the default number of matched frames, and α is the weight parameter.

$$L_{conf}(x,c)=-\sum_{i\in Pos} x_{ij}\log\left(\hat{c}_i\right)-\sum_{i\in Neg}\log\left(1-\hat{c}_i\right) \tag{2}$$

$$L_{loc}(x,l,g)=\sum_{i\in Pos} x_{ij}\sum_{m\in cx,cy,w,h} smooth_{L1}\left(l_i^m-g_i^m\right) \tag{3}$$

where x_{ij} denotes the matching result, if the default frame i matches the real target frame j, then $x_{ij}=1$, otherwise it is 0; $\hat{c}_i$ denotes the predicted value of the category confidence of the default frame i; l_i is the positional offset of the predicted frames; g_i is the positional offset of the real frames; *Pos* denotes the set of default boxes that match the true target boxes, i.e., the set of positive samples (the prediction results of these boxes are used to compute the positive sample portion of the categorical loss); *Neg* denotes the set of default boxes that do not match any of the true target boxes, i.e., the set of negative samples (the prediction results of these boxes are used to compute the negative sample portion of the categorical loss).

When performing target detection, SSD generates a large number of prediction frames through multi-scale feature maps and performs non-maximum suppression (NMS) on these prediction frames, retaining the frames with the highest level of confidence as the final detection results. SSD's multi-scale feature maps and the default frame mechanism enable it to efficiently detect targets at different scales.

2.2 SENet attention mechanisms

SENet is a module that enhances the representational capabilities of convolutional neural networks by explicitly modelling the dependencies between feature channels (Monfort and Senet, 2020). SENet performs well in several computer vision tasks, especially in image classification, target detection tasks such as image classification, target detection, etc., and the attention mechanism it introduces can significantly improve the performance of the network. In this paper, we will introduce the basic principle, network structure, and application of SENet to improve the SSD model.

SENet recalibrates the channel weights of the feature map through a simple but effective squeeze and excitation operation, which enhances useful features and suppresses useless ones. The mechanism consists of three steps: squeeze, excitation and reweight.

- *Squeeze:* captures global spatial information by compressing the spatial dimensions of each feature channel into a global description through the global average pooling operation.

$$z_c = F_{sq}\left(u_c\right) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j) \tag{4}$$

  where z_c denotes the global description of the c^{th} channel, $u_c(i, j)$ denotes the eigenvalue of the c^{th} channel at position (i, j), and H and W are the height and width of the eigenmap, respectively.

- *Excitation:* captures the nonlinear interdependencies between feature channels through two fully connected layers and an activation function. The number of feature channels is first reduced and then the original channel count is restored to reduce the number of parameters and computations.

$$s_c = F_{ex}(z, W) = \sigma\left(W_2 \delta\left(W_1 z\right)\right) \tag{5}$$

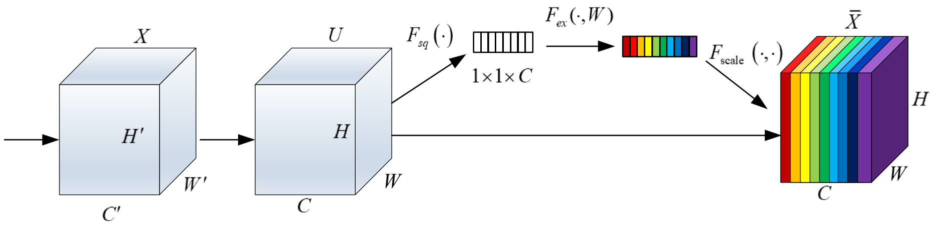
  where s_c denotes the excitation value of the c^{th} channel, W_1 and W_2 are the weight matrices of the fully connected layer, δ denotes the ReLU activation function, and σ denotes the sigmoid activation function.

- *Reweight:* completes the recalibration of the feature map by applying excitation value to original feature map by multiplication by channel.

$$\hat{u}_c = s_c \cdot u_c \tag{6}$$

  where $\hat{u}_c$ denotes the c^{th} channel feature map after recalibration. The specific structure of the SENet module is shown in Figure 1.

Figure 1 Structure of SENet (see online version for colours)



2.3 GhostNet

GhostNet is a lightweight neural network architecture proposed by Huawei, which effectively reduces the amount of computation and the number of parameters by introducing the Ghost module, while maintaining the expressiveness of the model. The core idea of GhostNet is to reduce the redundant feature maps, which speeds up the computation and improves the efficiency.

The GhostNet framework utilises the Ghost module to produce an increased number of feature maps, ensuring that the network retains its high level of expressiveness while simultaneously minimising computational demands. This module is comprised of two distinct components: the principal convolution layer that creates the foundational feature map and the linear transformation component that generates the auxiliary Ghost feature map. First, the base feature map is generated by standard convolution operation. This step is mainly used to extract the basic information of the input features.

$$Y_{primary} = X * W + b \tag{7}$$

where X denotes the input feature map, W denotes the convolution kernel, b denotes the bias, and $*$ denotes the convolution operation.

Next, a series of simple linear operations (e.g., point-by-point convolution, element-by-element operations, etc.) are performed to generate more feature maps, which are called Ghost feature maps.

$$Y_{ghost} = \sum_{i=1}^{k} \alpha_i \odot Y_{primary} \tag{8}$$

where Y_{ghost} denotes the generated Ghost feature maps, α_i denotes the different linear transformation weights, $\odot$ denotes element-by-element operation, and k is the number of generated Ghost feature maps.

Finally, the base feature map and the Ghost feature map are combined together to form the final output feature map.

$$Y_{output} = \left[Y_{primary}, Y_{ghost}\right] \tag{9}$$

where [,] denotes the feature map stitching operation.

By introducing the Ghost module, GhostNet significantly reduces the amount of computation and the number of parameters while guaranteeing network performance.

3 Target recognition based on improved SSD

3.1 Sorting target feature extraction

Target feature extraction is the first step of target recognition, which determines whether the model can accurately and efficiently recognise sorting targets. The improved SSD model mainly utilises the improved GhostNet network in the feature extraction phase, combined with the SE-Net module, in order to improve the efficiency and accuracy of feature extraction.

In order to further enhance the performance of the target detection network, this study makes three improvements to the original GhostNet. These improvements mainly target the computational efficiency and feature extraction capability of the network to ensure the accuracy of the detection while increasing the detection rate.

3.1.1 Introduction of dynamic convolution.

Dynamic convolution is a technique that adaptively adjusts the convolution kernel according to the input features (Jiang et al., 2020), which can improve the network's ability to adapt to different features. In traditional convolution operation, the convolution kernel is fixed, and this static approach may show inadequacy in the face of diverse inputs.

In the improved GhostNet, multiple convolutional kernels are learnt as shown as follows:

$$Y_{primary}^{i} = X * W_i + b_i \quad (i = 1, 2, \ldots, K) \tag{10}$$

where W_i and b_i denote the i^{th} convolutional kernel and its corresponding bias, respectively; K is the number of convolutional kernels.

Adaptive combination of convolutional kernels based on input features is shown as follows:

$$Y_{dynamic} = \sum_{i=1}^{K} \alpha_i \cdot Y_{primary}^{i} \tag{11}$$

where α_i is the weights learned adaptively based on the input features, $\sum_{i=1}^{K} \alpha_i = 1$, $Y_{dyanmic}$ is the feature map after dynamic combination.

With the introduction of dynamic convolution, GhostNet is able to adaptively adjust the convolution operation according to different input features, which makes the feature extraction more accurate and improves precision and robustness.

3.1.2 Enhanced feature fusion

In order to further enhance the diversity and information of feature extraction, the improved GhostNet is optimised in terms of feature fusion. Specifically, we introduce a multi-level feature fusion method to effectively fuse features at different levels to make full use of multi-scale feature information.

The process of multilevel feature extraction is shown below; the

$$F_l = GhostConv_l(X) \tag{12}$$

where F_l denotes the feature map of the l^{th} layer and $GhostConv_l$ denotes the convolution operation of the l^{th} layer of GhostNet.

The feature fusion is done in the following manner.

$$F_{fused} = \sum_{l=1}^{L} \beta_l \cdot F_l \tag{13}$$

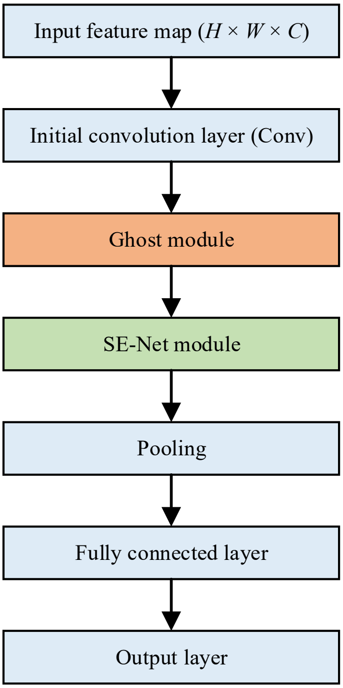
where β_l is the fusion weight of the l^{th} layer feature map, L is the number of layers to be fused, and F_{fused} is the fused feature map.

Through multilevel feature fusion, the network is able to utilise different levels of feature information at the same time, thus improving feature expressiveness and detection accuracy.

3.1.3 Inclusion of SE-Net module

The squeeze-and-excitation networks (SE-Net) module is introduced in GhostNet. The SE-Net module enhances the representation capability of the network by explicitly modelling the dependencies between feature channels, enhancing the focus on important features and suppressing useless features.

Figure 2 Improved GhostNet structure (see online version for colours)


Input feature map (H × W × C)
Initial convolution layer (Conv)
Ghost module
SE-Net module
Pooling
Fully connected layer
Output layer

The integration of the SE-Net module in GhostNet can be achieved by adding the SE module after the output of the Ghost module. The Ghost module first generates the base feature map and the Ghost feature map by standard convolution and linear transformation, and combines them into the final output feature map as shown in equation (8). Then, the output feature maps of the Ghost module are used as inputs to the SE-Net module, and the feature maps are recalibrated by the SE-Net module.

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} Y_{output,c}(i, j) \tag{14}$$

$$\hat{Y}_{output,c} = s_c \cdot Y_{output,c} \tag{15}$$

where $\hat{Y}_{output,c}$ denotes the c^{th} channel feature map after recalibration by the SE-Net module.

Eventually, the improved GhostNet structure is shown in Figure 2. By introducing the SE-Net module into GhostNet, the network is able to capture important features more effectively and suppress useless features, thus improving the accuracy of target detection.

During feature extraction, the SE-Net module enhances the focus on important features and suppresses useless features by explicitly modelling the dependencies between feature channels. In this study, we improved the SE-Net module in order to further enhance the feature extraction capability and target detection accuracy of GhostNet. The traditional SE-Net module uses ReLU and sigmoid activation functions, which perform well in terms of feature extraction and representation capabilities, but may suffer from high computational complexity in some application scenarios. To improve the computational efficiency, we replace the ReLU and sigmoid activation functions with lightweight h-swish activation functions (Huang et al., 2023).

The h-swish activation function is defined as follows:

$$h\text{-}swish(x) = x \cdot \frac{ReLU6(x+3)}{6} \tag{16}$$

where ReLU6 is the ReLU activation function restricted between 0 and 6:

$$ReLU6(x) = \min(\max(0, x), 6) \tag{17}$$

By introducing the *h-swish* activation function, the computational effort can be effectively reduced while maintaining or improving the model performance. Specifically, the h-swish activation function is applied to excitation in the SE-Net module, i.e., equation (5) changes to equation (18).

$$s_c = \sigma\left(W_2 \cdot h\text{-}swish\left(W_1 \cdot z\right)\right) \tag{18}$$

Through these steps, the improved GhostNet and SE-Net modules work in tandem to ensure that the extracted feature maps can be better used for target recognition tasks.

3.2 Sorting target types and spatial parameters

After completing target feature extraction, the next step is to recognise the target type and determine the spatial parameters. These steps determine how efficiently the industrial robot can perform the sorting task. With the improved GhostNet and SE-Net modules, the improved SSD model performs well in classification and regression tasks, accurately identifying the type of sorting targets and determining their spatial parameters. The classification module calculates the class probability of each candidate box using convolution and softmax activation functions, while the regression module predicts the offsets of spatial parameters through convolution operations and applies them to the default boxes to calculate the final spatial parameters.

The classification module uses the extracted feature maps to generate the category confidence of each candidate box using convolutional operations. Firstly, the feature maps output from the improved GhostNet module are used as input and a series of convolution operations are used to generate the category confidence maps.

$$C_{i,j,k} = Conv\left(F_{i,j,k}\right) \tag{19}$$

where $C_{i,j,k}$ denotes the confidence of the i^{th} candidate frame in the k^{th} category on the j^{th} feature map, *Conv* denotes the convolution operation, and $F_{i,j,k}$ denotes the feature map.

The softmax activation function is then applied to the generated category confidence maps to calculate the probability of each category.

$$P_{i,j,k} = \frac{\exp\left(C_{i,j,k}\right)}{\sum_{k=1}^{K}\exp\left(C_{i,j,k}\right)} \tag{20}$$

where $P_{i,j,k}$ denotes the probability that the i^{th} candidate frame is in the k^{th} category on the j^{th} feature map.

Through the above steps, the classification module is able to accurately identify the target type of each candidate box.

After determining the type of sorting target, the next step is to determine its spatial parameters, including the centre coordinates, width and height of the target. The determination of the spatial parameters is achieved through the regression module in the SSD model.

The regression module predicts the spatial parameters of each candidate frame by performing convolution operations on the feature map. Firstly, the feature map output from the improved GhostNet module is taken as input and the predicted spatial parameters of each candidate box are generated by a series of convolution operations.

$$\Delta x_{i,j}, \Delta y_{i,j}, \Delta w_{i,j}, \Delta h_{i,j} = Conv\left(F_{i,j,k}\right) \tag{21}$$

where $\Delta x_{i,j}$, $\Delta y_{i,j}$, $\Delta w_{i,j}$, $\Delta h_{i,j}$ denote the offset of the centre coordinates and dimensions of the i^{th} candidate frame on the j^{th} feature map, respectively.

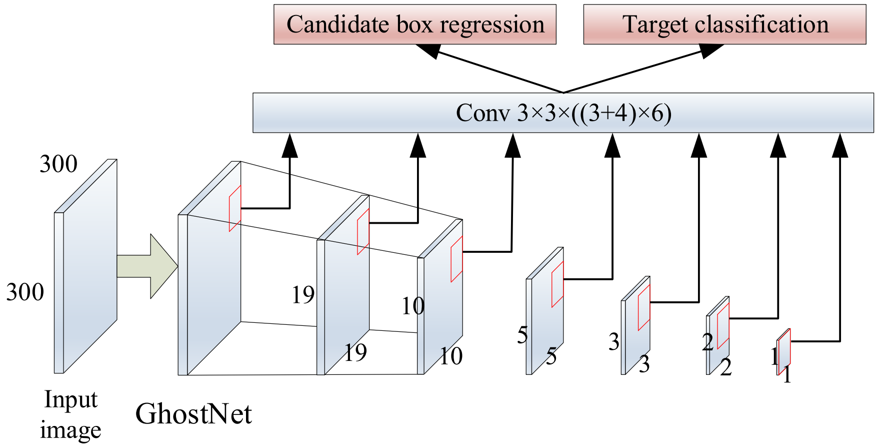
Apply the offset to the default box to calculate the final spatial parameters.

$$\begin{aligned} x_i &= x_i^{default} + \Delta x_{i,j} \cdot w_i^{default} \\ y_i &= y_i^{default} + \Delta y_{i,j} \cdot h_i^{default} \\ w_i &= w_i^{default} \cdot \exp\left(\Delta w_{i,j}\right) \\ h_i &= h_i^{default} \cdot \exp\left(\Delta h_{i,j}\right) \end{aligned} \tag{22}$$

where $x_i^{default}$, $y_i^{default}$, $w_i^{default}$, $h_i^{default}$ denote the centre coordinates and dimensions of the i^{th} default box respectively.

Through the above steps, the regression module can accurately determine the spatial parameters of each candidate frame. Finally, the network structure of GhostNet-SSD is shown in Figure 3.

Figure 3 Network structure of GhostNet-SSD (see online version for colours)



4 Robot zero position calibration

Robot zero calibration is a key step in achieving accurate motion control, especially in vision-based sorting control systems, where accurate calibration ensures that the robot accurately identifies and locates the target object. In this paper, the least squares method is used to implement vision-based calibration, aiming to establish the mapping relationship between the robot coordinate system and the vision coordinate system through accurate mathematical models and algorithms, so as to improve the accuracy and stability of the system.

The core task of robot zero calibration is to determine the translation relationship between the vision system's coordinate system and the robot's coordinate system. This relationship is usually established through a series of known calibration points. Setting the coordinates of the calibration points in the vision coordinate system as (X_i, Y_i), which corresponds to (x_i, y_i) in the robot coordinate system, it is necessary to find a transformation matrix T such that the points in the vision coordinate system can be accurately mapped to the robot coordinate system after the transformation.

In order to establish the transformation relationship between the visual coordinate system and the robot coordinate system, a linear transformation model can be used. In general, the transformation relation can be expressed as follows.

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = T \cdot \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \tag{23}$$

where T is a 2 × 3 transformation matrix.

$$T = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \tag{24}$$

The transformation relationship can be specifically developed as follows.

$$x_i = aX_i + bY_i + c \tag{25}$$

$$y_i = dX_i + eY_i + f \tag{26}$$

In order to solve for the parameters *a*, *b*, *c*, *d*, *e*, *f* in the transformation matrix *T*, we can use the least squares method.

$$\min_{a,b,c,d,e,f} \sum_{i=1}^{n} \left((aX_i + bY_i + c - x_i)^2 + (dX_i + eY_i + f - y_i)^2 \right) \tag{27}$$

The above objective function is a typical least squares problem that can be computed in a simplified way by using the matrix form. Setting:

$$A = \begin{bmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & X_2 & Y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & X_n & Y_n & 1 \end{bmatrix} \tag{28}$$

$$B = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_n \\ y_n \end{bmatrix} \tag{29}$$

$$\theta = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \tag{30}$$

The least squares problem can then be expressed as solving a system of linear equations $A\theta = B$. By solving the above system of equations, the parameter θ in the transformation matrix *T* can be obtained. In order to solve this system of equations, we can use the standard solution of the least squares method.

$$\theta = \left(A^T A \right)^{-1} A^T B \tag{31}$$

Finally, the parameter θ in the transformation matrix T is calculated using the least squares formula. Using the solved parameter θ, the transformation relationship between the visual coordinate system and the robot coordinate system is established.

5 Sorting control for industrial robots

After the target recognition by the improved SSD model, it is necessary to carry out path planning for the recognised targets, and finally drive the robot to complete the sorting task by the motion control algorithm. The sorting control model mainly includes the following steps:

1. target identification: the improved SSD model is used to identify the sorting target and determine its type and spatial parameters (centre coordinates x_i, y_i and dimensions w_i, h_i)
2. path planning: based on the spatial parameters of the target and the robot's workspace, path planning is carried out to ensure that the robot can efficiently and accurately reach the target position
3. motion control: based on the planned path, the motion control algorithm is used to drive the robot to complete the sorting action.

Path planning is one of the key steps in sorting control. The purpose of path planning is to plan an optimal path under the premise of ensuring sorting efficiency and accuracy, so that the robot can reach the target location smoothly. In practical applications, commonly used path planning algorithms include A* algorithm, Dijkstra algorithm and rapid search random tree (RRT) algorithm. In this paper, the RRT algorithm is used for path planning, and the specific steps are as follows:

Step 1 Initialise a tree T at the start position $\mathbf{P}_{start}$.

Step 2 Sample a random point $\mathbf{P}_{rand}$ in the workspace.

Step 3 Find the node $\mathbf{P}_{rand}$ that is closest to $\mathbf{P}_{nearest}$ in the tree and extend it by one step from $\mathbf{P}_{nearest}$ towards $\mathbf{P}_{rand}$ to generate the new node $\mathbf{P}_{new}$.

Step 4 Add $\mathbf{P}_{new}$ to the tree T and record the path.

Step 5 If $\mathbf{P}_{new}$ is close enough to the goal location $\mathbf{P}_{goal}$, the path planning is successful and the path $\mathbf{P}$ is generated.

The above steps are repeated until a path is found from $\mathbf{P}_{start}$ to $\mathbf{P}_{goal}$.

Motion control is the final aspect of sorting control, which is achieved by controlling the robot's joints and end-effector to achieve target grasping and sorting. In this paper, motion control method based on PID controller is used. Robot motion control requires the establishment of a kinematic model of the robot. For a robot with n degrees of freedom, the position and attitude of its end-effector can be expressed as:

$$\mathbf{T} = \mathbf{T}_0 \cdot \prod_{i=1}^{n} \mathbf{T}_i\left(\theta_i\right) \tag{32}$$

where $\mathbf{T}_0$ is the base coordinate system, $\mathbf{T}_i(\theta_i)$ is the transformation matrix of the i^{th} joint, and θ_i is the angle of the i^{th} joint.

PID controller is a commonly used closed-loop controller to achieve precise control of the target position through proportional (P), integral (I) and differential (D) control. The output of the PID controller can be expressed as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \tag{33}$$

where $u(t)$ is the control output; $e(t)$ is the error; K_p, K_i and K_d are the proportional, integral and differential gains respectively. The robot can accurately reach the target position and complete the sorting task through the PID controller.

6 Experimental results and analyses

We validate the improved SSD model and its application in industrial robot sorting control system through a series of experiments. The experiments are mainly divided into target recognition performance tests and sorting control accuracy tests. Through these experiments, the effectiveness and reliability of the system were verified.

The test dataset used in this study contains several common industrial sorting targets, aiming at simulating the real industrial environment. These datasets are collected from the actual production line and pretreated to ensure the quality and diversity of the data. The dataset includes the changes of different lighting conditions, background complexity and the shape of the target object to verify the performance of the improved SSD model in different scenes. To assess the effectiveness of the improved sorting control system, a series of typical sorting tasks were set up, as shown in Table 1.

Target recognition is the foundation of a sorting control system, and its accuracy and real-time performance directly affect the performance of the entire system. We conducted recognition tests on several typical sorting targets using an improved SSD model. The test dataset was selected to contain image datasets of several common industrial sorting targets. Mean accuracy (mAP) and detection speed (FPS) were used as evaluation metrics (Zhou et al., 2020).

Table 1 Industrial robot sorting control task list

Task no.	*Mission statement*	*Target type*	*Target number*	*Workspace*	*Hindrance*
1	Fixed position static spreading exercise	Casts	10	1 m × 1 m	Not have
2	Dynamic splitting of moving targets	Bottles	5	1 m × 1.5 m	There are
3	Mixed-objective complex breakout	Multibody	20	1.5 m × 2 m	There are
4	High-speed dynamic target separation	Ball game	25	2 m × 2 m	Not have
5	Multi-robot collaboration	Bottles	30	2 m × 2.5 m	There are

mAP is a comprehensive evaluation index to measure the overall accuracy of the target detection algorithm. It is obtained by calculating the average precision and recall under different confidence thresholds. The higher mAP indicates that the model has better recognition ability.

FPS is used to measure the real-time performance of the system. It is an important index to evaluate the processing speed of target detection algorithm. A higher FPS means that the system can process images faster and is suitable for real-time applications.

The improved SSD model is compared with the original SSD model. As shown in Table 2, the improved SSD model outperforms the original SSD model in both accuracy and real-time of target recognition.

Table 2 Performance comparison between improved SSD and original SSD

Model	*mAP (%)*	*FPS (frames per second)*
Original SSD	74.3	22.5
Improved SSD	81.7	28.9

It can be seen that the improved SSD model improves the average accuracy by 7.4 % and the detection speed by 28.4 %. In order to verify the accuracy of the sorting control, actual robot sorting tests were performed, including static and dynamic sorting tasks. According to the task descriptions in Table 1, static sorting at a fixed position and dynamic sorting tests with moving targets were performed. The evaluation indexes are sorting success rate and error distance. The experimental results are shown in Table 3, which shows that the improved sorting control system has a high success rate in both static and dynamic tasks, and the error distance is within the acceptable range.

Table 3 Accuracy test results of sorting control system

Type of mission	*Success rate of sub-exercise (%)*	*Tolerance distance (cm)*
Static sorting	98.5	0.4
Dynamic sorting	95.7	0.6

It can be seen that the improved sorting control system shows good sorting accuracy with a success rate of 98.5% in the static task and 95.7% in the dynamic task with error distances of 0.4 cm and 0.6 cm, respectively.

On the whole, the improved SSD model has a marked enhancement in both target recognition accuracy and real-time performance, and the improved sorting control system has a high sorting success rate and accuracy in practical applications. These experimental results verify the effectiveness and practicality of the improved method, and prove the application prospect of the system in industrial robot vision sorting control.

7 Conclusions

In this paper, an improved SSD-based vision sorting control system for industrial robots is proposed, which effectively solves the limitations of traditional methods in dealing with complex backgrounds and diverse target objects. By replacing the VGG-16 backbone network of the traditional SSD algorithm with a GhostNet network, the detection rate is significantly improved. The SENet attention mechanism is introduced to ensure the detection accuracy. In addition, GhostNet is improved to enhance the flexibility and effectiveness of feature extraction by introducing dynamic convolution technique. The SENet module is optimised with a more efficient activation function, which further improves the model performance. The following conclusions can be drawn from the experiments on a variety of industrial sorting tasks:

1 replacing the VGG-16 backbone network with the GhostNet network significantly increases the detection rate and enhances the real-time performance of the system

2 the introduction of the SENet attention mechanism improves the precision of the target detection network, especially performing well in complex contexts

3 the improved GhostNet improves the flexibility and effectiveness of feature extraction by dynamic convolution technique

4 the optimised SENet module reduces the computational complexity and further improves the model performance.

The experimental data in this paper, mainly from a variety of industrial sorting tasks, validates the effectiveness and usefulness of the improved approach. However, the limitations of the dataset may affect the generalisation ability of the model in different industrial environments. Future work should consider introducing more datasets from different industrial domains to validate the effectiveness of the model in a wider range of application scenarios.

Acknowledgements

This work is supported by the Science and Technology Research Project of Chongqing Education Commission (No. KJQN202201905), the Chongqing Institute of Engineering Research Project (No. 2021xzky05), and the 2022 College Student Innovation and Entrepreneurship Training Program Project (No. 202212608005).

References

Abbood, W.T., Abdullah, O.I. and Khalid, E.A. (2020) ‘A real-time automated sorting of robotic vision system based on the interactive design approach’, *International Journal on Interactive Design and Manufacturing (IJIDeM)*, Vol. 14, No. 1, pp.201–209.

Charmette, B., Royer, E. and Chausse, F. (2016) ‘Vision-based robot localization based on the efficient matching of planar features’, *Machine Vision and Applications*, Vol. 27, pp.415–436.

Ciora, R.A. and Simion, C.M. (2014) ‘Industrial applications of image processing’, *Acta Universitatis Cibiniensis. Technical Series*, Vol. 64, No. 1, pp.17–21.

Huang, T., Zhu, J., Liu, Y. et al. (2023) ‘UAV aerial image target detection based on BLUR-YOLO’, *Remote Sensing Letters*, Vol. 14, No. 2, pp.186–196.

Jiang, Z.-H., Yu, W., Zhou, D. et al. (2020) ‘ConvBERT: improving BERT with span-based dynamic convolution’, *Advances in Neural Information Processing Systems*, Vol. 33, pp.12837–12848.

Kiyokawa, T., Takamatsu, J. and Koyanaka, S. (2022) ‘Challenges for future robotic sorters of mixed industrial waste: a survey’, *IEEE Transactions on Automation Science and Engineering*, Vol. 21, No. 1, pp.1023–1040.

Li, S., Sultonov, F., Tursunboev, J. et al. (2022) 'Ghostformer: a GhostNet-based two-stage transformer for small object detection', *Sensors*, Vol. 22, No. 18, pp.6939.

Lin, Y., Zhou, H., Chen, M. et al. (2019) 'Automatic sorting system for industrial robot with 3D visual perception and natural language interaction', *Measurement and Control*, Vol. 52, Nos. 1–2, pp.100–115.

Liu, P., Zhang, X. and Xu, Z. (2024) 'Yolov4 algorithm for target detection in existing intelligent waste sorting systems', *Highlights in Science, Engineering and Technology*, Vol. 81, pp.237–242.

Ma, W., Wang, X. and Yu, J. (2020) 'A lightweight feature fusion single shot multibox detector for garbage detection', *IEEE Access*, Vol. 8, pp.188577–188586.

Monfort, J-B. and Senet, P. (2020) 'Leg ulcers in sickle-cell disease: treatment update', *Advances in Wound Care*, Vol. 9, No. 6, pp.348–356.

Paoletti, M.E., Haut, J.M., Pereira, N.S. et al. (2021) 'Ghostnet for hyperspectral image classification', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 59, No. 12, pp.10378–10393.

Shaikat, A.S., Akter, S. and Salma, U. (2020) 'Computer vision based industrial robotic arm for sorting objects by color and height', *Journal of Engineering Advancements*, Vol. 1, No. 4, pp.116–122.

Vayda, A. and Kak, A.C. (1991) 'A robot vision system for recognition of generic shaped objects', *CVGIP: Image Understanding*, Vol. 54, No. 1, pp.1–46.

Wang, T., Yao, Y., Chen, Y. et al. (2018) 'Auto-sorting system toward smart factory based on deep learning for image segmentation', *IEEE Sensors Journal*, Vol. 18, No. 20, pp.8493–8501.

Wang, Y., Hong, K., Zou, J. et al. (2019) 'A CNN-based visual sorting system with cloud-edge computing for flexible manufacturing systems', *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 7, pp.4726–4735.

Wu, X., Ling, X. and Liu, J. (2019) 'Location recognition algorithm for vision-based industrial sorting robot via deep learning', *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 33, No. 7, pp.1955009.

Yan, C., Zhang, H., Li, X. et al. (2022) 'R-SSD: refined single shot multibox detector for pedestrian detection', *Applied Intelligence*, Vol. 52, No. 9, pp.10430–10447.

Zhou, K., Meng, Z., He, M. et al. (2020) 'Design and test of a sorting device based on machine vision', *IEEE Access*, Vol. 8, pp.27178–27187.