Journal of Intelligent & Robotic Systems (2022) 104: 77
https://doi.org/10.1007/s10846-021-01564-2

SHORT PAPER

MGBM-YOLO: a Faster Light-Weight Object Detection Model for Robotic Grasping of Bolster Spring Based on Image-Based Visual Servoing

Huanlong Liu[1,2] · Dafa Li[1,2] · Bin Jiang[1,3] · Jianyi Zhou[1,2] · Tao Wei[1,2] · Xinliang Yao[1,2]

Received: 1 July 2021 / Accepted: 19 December 2021 / Published online: 14 April 2022

Abstract
The rapid detection and accurate positioning of bolster spring with complex geometric features in cluttered background is highly important for the grasping task of bolster spring in overhauling workshop. To achieve a better trade-off among positioning accuracy, running time and model size, two MobileNetv3 and GhostNet-based modified YOLOv3 (MGBM-YOLO) models are proposed in this paper: MGBM-YOLO-Large and MGBM-YOLO-Small, which are applied to the robotic grasping system of bolster spring based on image-based visual servoing (IBVS). The proposed MGBM-YOLO models are trained and evaluated on Pascal VOC2007 and Pascal VOC2012 datasets. The results show that compared with the original YOLOv3 model based on Darknet53, the sizes of the proposed Large and Small models are reduced by 59% and 66% respectively, and the model parameters are reduced by 63% and 66% respectively. The detection speed is 4× and 5.5× faster respectively when the difference of mAP is small. To improve the convergence speed of IBVS system, an online depth estimation method based on the area feature of detection bounding box of bolster spring is proposed. The comparative experiments of visual servoing grasping are conducted on the robotic grasping platform of bolster spring. The experimental results show that the two MGBM-YOLO detection models proposed in this paper can meet the requirements of fast detection and positioning of bolster spring robotic grasping system based on IBVS. The proposed method of feature depth online estimation is of great significance to accelerate the convergence speed of visual servoing system.

Keywords Modified YOLOv3 model · Light-weight neural network · Depth online estimation · Visual servoing · Bolster spring grasping

1 Introduction

Railway freight transportation is one of the significant ways of freight transportation in China, and its strict operation security requirements are always the first priority of the transportation system. Bolster spring, as one of the key components of railway freight trains to transmit load and mitigate vibration and impact, needs to be overhauled regularly [1]. During the overhauling process, the bolster spring needs to be removed from the side frame of the bogie, and then installed back to its original position after the inspection is completed. Therefore, it is of great significance to carry out the research on intelligent detection, positioning and grasping of bolster spring.

At present, the handling of bolster spring is completed by manual operation, which has the problems of high labor intensity and low operation efficiency. The rapid development of industrial robots and visual servoing technology has made it possible to apply visual servoing-based robotic grasping technology to the automatic positioning and grasping of bogie bolster spring, thereby replacing manual operations to reduce labor costs and improve operation efficiency.

✉ Dafa Li
lidafa@my.swjtu.edu.cn

[1] Engineering Research Center of Advanced Driving Energy-Saving Technology, Ministry of Education, Chengdu 610031, People's Republic of China

[2] School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, People's Republic of China

[3] Graduate School of Tangshan, Southwest Jiaotong University, Tangshan 063000, People's Republic of China

According to the different constitution of control system, visual servoing is usually classified into two categories:]image-based visual servoing (IBVS) and position-based visual servoing (PBVS) [2, 3]. Compared with PBVS, IBVS]is robust to the camera model and hand-eye calibration error due to its direct closed-loop feedback control in image space [4]. In addition, IBVS does not necessitate estimating the pose of the object in Cartesian space, which reduces the complexity of the system.

Bolster spring has the characteristics of cluttered background, complex geometric features, and multi-scale changes, making its detection and positioning extremely challenging. The significant advantages of object detection algorithm based on deep learning in object detecting and positioning provide a new control idea and research entry point for the grasping of bogie bolster spring of the railway freight train. Compared with YOLOv1 and YOLOv2, the speed and accuracy of YOLOv3 object detection model are significantly improved, and the existence of multi-scale prediction mechanism also makes the algorithm have good adaptability for objects of different sizes. Therefore, in our previous works [5], the YOLOv3 object detection model was employed to detect and position the bolster spring. It was found that the original YOLOv3 model had too much FLOPs and too low detection speed on CPU, which was not conducive to meet the real-time requirements of the robotic visual servoing grasping system of bolster spring. At the same time, the feature depth in the image Jacobian matrix adopted was the depth value in the desired pose, which led to the slow convergence speed of the visual servoing system. Inspired by the latest light-weight network design of MobileNetv3 [6] and GhostNet [7], this paper modifies the backbone network of YOLOv3 by combining channel split and channel shuffle as well as GhostNet bottleneck layer and MobileNetv3 network architecture, and obtains two faster light-weight object detection models: MGBM-YOLO-Large and MGBM-YOLO-Small. In addition, in terms of feature depth estimation, a new depth-independent image Jacobian matrix is proposed for the design of IBVS controller based on the area feature of the object detection bounding box of bolster spring. Experiments conducted on the robotic grasping physical platform of bolster spring show that the proposed object detection models and feature depth estimation method have good performance. The main contributions of this paper are as follows:

(1) The MobileNetv3 network architecture and GhostNet bottleneck layer are analyzed in detail. Two modified YOLOv3 models, MGBM-YOLO-Large and MGBM-YOLO-Small, based on GhostNet bottleneck layer and MobileNetv3 network architecture, are proposed by combining channel split and channel shuffle. The sizes of the two MGBM-YOLO models are smaller and detection speeds are faster.
(2) The positioning accuracy and detection speeds of the two modified light-weight MGBM-YOLO models can meet the control requirements of the robotic grasping system of bolster spring. They are applied to the IBVS system based on the corner points features of the object detection bounding box to realize the automatic positioning and grasping of the bolster spring with complex geometric features in cluttered background.
(3) Based on the imaging characteristics of pinhole camera model, the relationship between image area feature and feature depth is analyzed, and an online image feature depth estimation method is proposed. The depth-independent image Jacobian matrix constructed by using the estimated depth value achieves a good effect of accelerating the convergence speed of visual servoing system.

The remaining of this paper is organized as follows. Section 2 provides a brief overview of related work. Section 3 introduces the MGBM-YOLO models in detail. A novel IBVS control system based on the corner points features of the MGBM-YOLO object detection bounding box is proposed in Section 4. Section 5 presents the experimental validation. The conclusion of this paper and the focus of future work are given in Section 6.

2 Related Work

2.1 Light-Weight Neural Network

The earliest research representative of light-weight networks is NIN [8], which adds Conv1 × 1 and Global Average Pooling (GAP) layers to the Conv3 × 3 layer. The number of parameters and FLOPs of Conv1 × 1 only account for 1/9 of Conv3 × 3, and the GAP layer does not contain parameters, so these two can effectively compress and accelerate the CNN model. Both GoogLeNet [9] and SqueezeNet [10] used Conv1 × 1 to replace Conv3 × 3 in large quantities and adopted the compression acceleration characteristics of Conv1 × 1 to efficiently reduce the number of model parameters and FLOPs. MobileNetv1 [11] proposed the Depthwise Separable Convolution (DwConv), through the combination of DwConv3 × 3 and Conv1 × 1 layers, greatly reducing the FLOPs of the network. However, the replacement of Conv3 × 3 by a large number of DwConv3 × 3 and Conv1 × 1 made Conv1 × 1 accounted for the FLOPs proportion of the overall network on the rise. To address the problem of the excessive proportion of Conv1 × 1, MobileNetv2 [12] proposed an inverted residual structure and linear bottlenecks to reduce the use of standard Conv1 × 1. ShuffleNetv1 [13]

employed Group Convolution (GConv) and channel shuffle to replace standard Conv1 × 1. In the latest studies, MobileNetv3 [6] introduces Network Architecture Search (NAS) in network design to optimize network architecture and parameters. ShuffleNetv2 [14] discarded the GConv1 × 1 in the ShuffleNetv1 by analyzing and comparing existing networks, using Concat and Channel Shuffle to reduce the FLOPs, and proposed four principles for light-weight network design. GhostNet [7] proposed the concept of redundant features, and then employed cheap linear operations to replace half of the convolution operations, reducing the number of model parameters and FLOPs.

2.2 Visual Servoing Based on Point Feature

The point feature-based visual servoing is the most widely applied method. To avoid the problem of multiple solutions and the singularity of the image Jacobian matrix, four coplanar feature points (and three of them are not collinear) are generally selected as the image features of visual servoing [15]. At present, most of the research literature on visual servoing based on feature points uses objects with simple geometric features and backgrounds as the research objects [16–21]. In [16], four black circular holes on a rectangle graphite tile were used as image feature points of visual servoing, and the visual servoing of a 10-DOF maintenance robot was simulated. In [17], four corners of a green rectangle were employed as image feature points to verify the effectiveness of the IBVS control method based on Eye-in-Hand monocular camera. In [18], in order to reduce feature interference and improve the accuracy of image processing, four low-noise and identical feature shapes were specially designed on A4 paper, which were used as image features for incremental visual servoing research. There are also many works that selected four corners of rectangular plastic block [19], four dots on square object [20], four black dots on A4 paper [21] as image features to carry out relevant IBVS research. A common point in the above studies is that objects with simple geometric features and backgrounds were adopted as the research objects of visual servoing to reduce the difficulty of image processing and feature point extraction. Few works have conducted research on visual servoing based on image points features for objects with complex geometric features and cluttered backgrounds.

3 MobileNetv3 and GhostNet-Based Modified YOLOv3 Model

In this section, firstly, based on the network architecture of MobileNetv3 and GhostNet, the MGBM-YOLO models are described. Then, the proposed MGBM-YOLO models and the original YOLOv3 model are trained and evaluated on the public datasets Pascal VOC2007 and Pascal VOC2012. Finally, the training on the custom bolster spring dataset is performed, and the trained detection model is used for the real-time detecting and positioning of the bolster spring in the visual servoing process.

3.1 MobileNetv3 and GhostNet Networks

As the latest light-weight neural network architecture, MobileNetv3 and GhostNet, respectively released by Google and Huawei, have greatly improved the accuracy and detection speed compared with MobileNetv1 and MobileNetv2. Based on the combination of complementary search techniques and novel architecture design, MobileNetv3 proposes two new MobileNet models: MobileNetv3-Large and MobileNetv3-Small [6]. The backbone network architectures are shown in Fig. 1.

As can be seen from Fig. 1, when the resolution of input image is $224^2 \times 3$, both MobileNetv3-Large and MobileNetv3-Small models downsample the input image five times with stride = 2 and finally output the feature map of $7^2 \times C$ (C denotes the number of channels). Through in-depth analysis, it can be found that the rapid increase of the network depth of MobileNetv3 is concentrated in the resolution stage of 14 × 14. In this stage, a large number of bottleneck layer modules are piled up in the network, so that the number of parameters and FLOPs are the highest in the whole network. The bottleneck layer of MobileNetv3 is shown in Fig. 2.

The h-swish activation function is introduced into the bottleneck layer of MobileNetv3, which effectively improves the accuracy of the network, but a large number of convolution operations and the nonlinear activation function make the FLOPs still large. It is considered unnecessary to employ a large number of parameters and FLOPs to generate redundant feature maps. Therefore, to reduce the number of model parameters, Ghost module adopts a large number of cheap linear operations instead of convolution operations to output some "ghost" feature maps of the intrinsic feature maps [7].

As shown in Fig. 3, the GhostNet bottleneck layer is composed of two stacked Ghost modules. The only difference is that a DwConv3 × 3 is introduced between the two Ghost modules of the GhostNet bottleneck layer with stride = 2 as the downsampling layer. Based on the network architecture of MobileNetv3, a more efficient and light-weight convolutional neural network, GhostNet, is obtained by replacing the bottleneck layer of MobileNetv3 by that of GhostNet.

3.2 Modified YOLOv3 Model

As the representative of one-stage object detection algorithm based on bounding box regression, YOLO series algorithms are widely used in object detection in complex backgrounds. In [22], the DR-YOLO model based on CNN is employed to

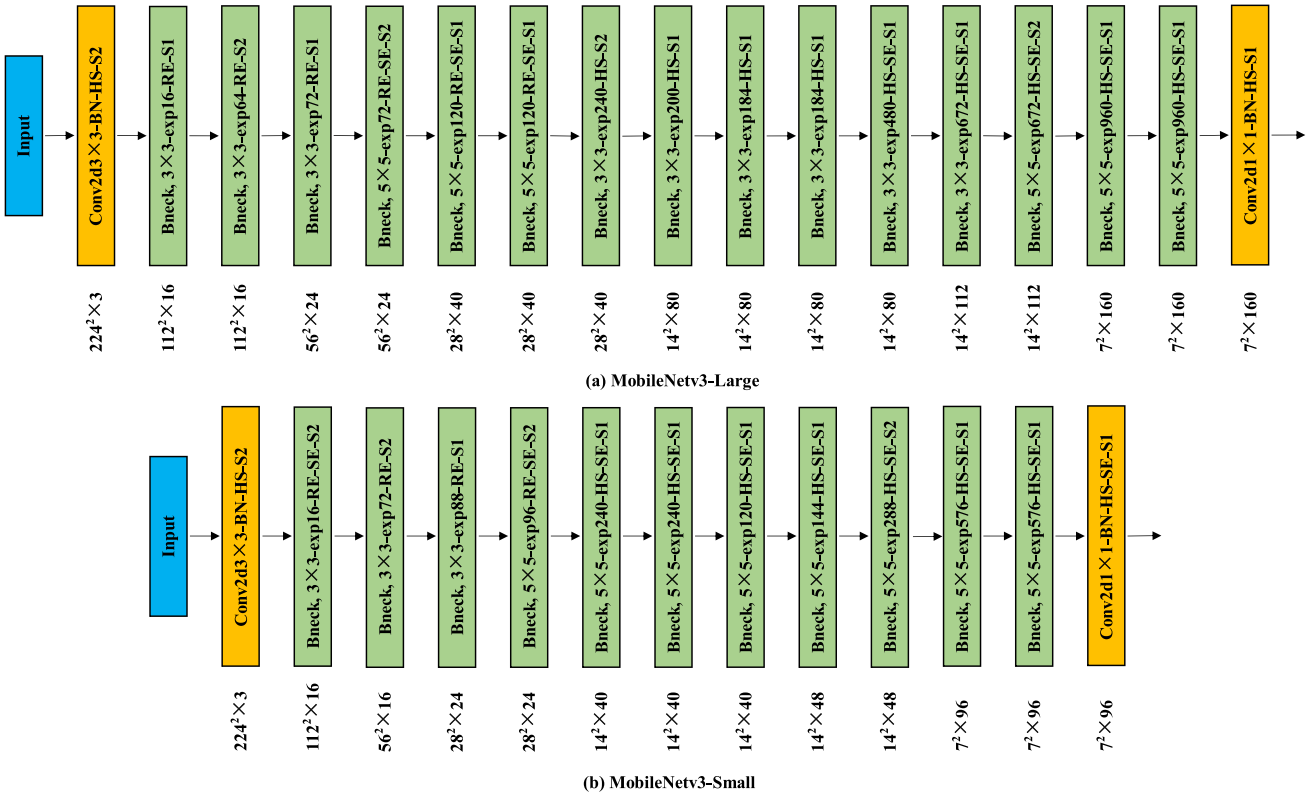

Fig. 1 Backbone of MobileNetv3: HS denotes h-swish; exp. denotes the number of channels after expansion; RE denotes ReLU; SE denotes Squeeze-and-Excitation module; S stands for stride

realize the coarse positioning and digital recognition of high-speed rail pole number under complicated background. In [23], the performance of various neural networks for traffic sign detection is evaluated, and it is found that YOLOv2, as the second-fastest detector, achieves very competitive accuracy results. Compared with YOLOv1 [24] and YOLOv2 [25], the detection accuracy and speed of YOLOv3 [26] are improved a lot, and it is favored by researchers because of its classic network architecture. YOLOv3 adopts Darknet53 as its backbone network, and the residual connection makes the network deepen rapidly. Its original network architecture is shown in Fig. 4.

It can be seen clearly from Fig. 4 that the original network architecture of YOLOv3 consists of four parts: input, backbone, neck and output. The backbone network aims to extract various features from the input image, then combine the corresponding features through the neck network, and finally perform multi-scale prediction of the object through the output part. When the resolution of input image is $416^2 \times 3$, the network outputs the prediction results of $13^2 \times 255$, $26^2 \times 255$, $52^2 \times 255$ scales after five times of downsampling with stripe = 2 and corresponding upsampling. The existence of a large number of residual blocks in the backbone network makes this part of the parameters account for a large proportion. Therefore, in order to lighten the model of YOLOv3, we consider improving it from the backbone network.

Inspired by the Ghost module in the GhostNet bottleneck layer using cheap linear operations instead of convolution operations to reduce the parameters and FLOPs, this paper proposes to introduce channel split in front of the Ghost module in the bottleneck layer of GhostNet, so as to divide a part of the tensor channel which was originally input into the Ghost module and pass it back directly. The modified bottleneck layer structure of GhostNet is shown in Fig. 5.

The two parts of the tensor obtained by channel split are concatenated using Concat after the Ghost module and the DwConv operation. Based on the significant advantages of channel shuffle in enhancing global feature information interaction and improving network expression ability, this paper introduces channel shuffle after tensor Concat to strengthen the mixing of information between channels, thereby improving the performance of the network. Five bottleneck layers with stride = 2 in the YOLOv3 backbone reduces the final resolution to 1/32 of the input, which is similar to the MobileNetv3 backbone architecture. Based on this, referring to the network framework of MobileNetv3, this paper adopts the modified GSbneck to replace the original bottleneck layer of MobileNetv3, and then the

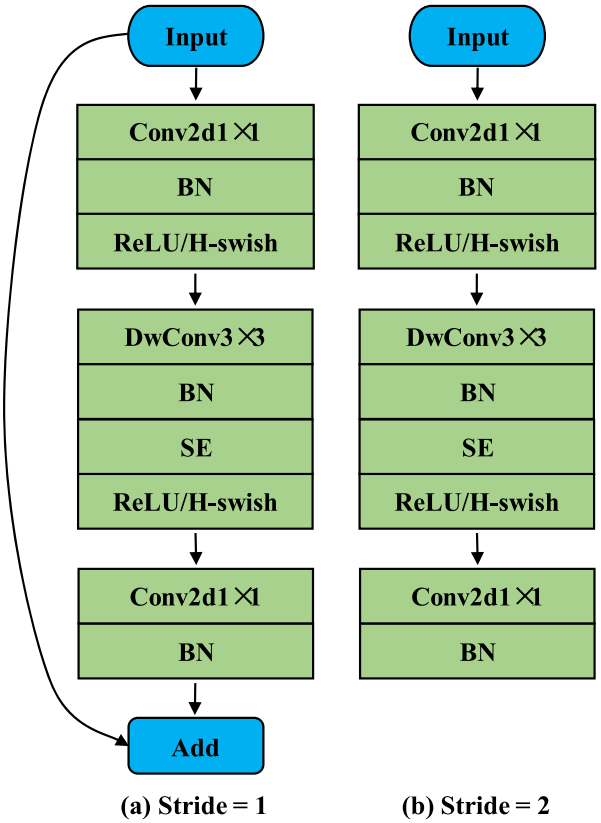

Fig. 2 MobileNetv3 bottleneck layer

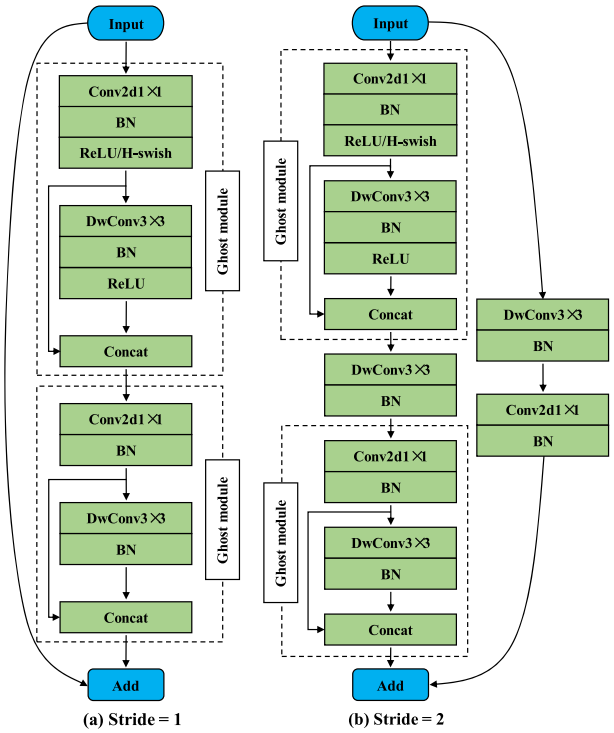

Fig. 3 GhostNet bottleneck layer

MGBM-YOLO-Large and MGBM-YOLO-Small models are obtained. The network architectures are shown in Fig. 6.

As shown in Fig. 6a and b, on the whole, the MGBM-YOLO network architectures retain the original network framework of YOLOv3 and still consist of four parts: input, backbone, neck and output. The main difference lies in the adoption of the Large and Small networks composed of the newly modified GSbneck module as the backbone, which constitutes the MGBM-YOLO-Large and MGBM-YOLO-Small models suitable for high resource utilization and low resource utilization respectively.

3.3 Comparison of Detection Performance on Pascal VOC Dataset

To validate the effectiveness of the proposed MGBM-YOLO object detection models, Pascal VOC2007 and Pascal VOC2012 are employed to train and evaluate the MGBM-YOLO-Large and MGBM-YOLO-Small models respectively. At the same time, in order to intuitively and clearly observe the performance changes before and after the modification of the models, the original YOLOv3 model is trained and evaluated on the same datasets. It should be noted that the training of the models is carried out on the GPU, while the evaluation is carried out on the CPU. Different image input resolutions (from 288 × 288 to 480 × 480) are set to obtain the accuracy and speed comparison between the MGBM-YOLO models and the original YOLOv3 model, as shown in Fig. 7.

At low resolution MGBM-YOLO-Small model achieves fast and fairly accurate results. With an input resolution of 288 × 288, the MGBM-YOLO-Small model can achieve a detection speed close to 9 FPS with comparable accuracy to the original YOLOv3 model. This is ideal for the rapid detection of industrial PC without GPU in the industrial field. At high resolution MGBM-YOLO-Large model can reach 0.840 mAP on VOC2007 while maintaining a good detection speed. To further compare the details of the proposed MGBM-YOLO models and the original YOLOv3 model, the comparison results under the input resolution of 416 × 416 and the hardware configuration.

information used are listed in Table 1.

It can be clearly seen from Table 1 that compared with the original YOLOv3 model, the mAP of the MGBM-YOLO-Large model proposed in this paper has increased significantly to 0.833; the mAP of the MGBM-YOLO-Small model has decreased slightly but the difference is small. However, at such an average accuracy level, the numbers of parameters of Large and Small models are reduced by about 63% and 66% respectively compared with that of the original YOLOv3 model, and the model sizes are also significantly reduced accordingly. When the input resolu-

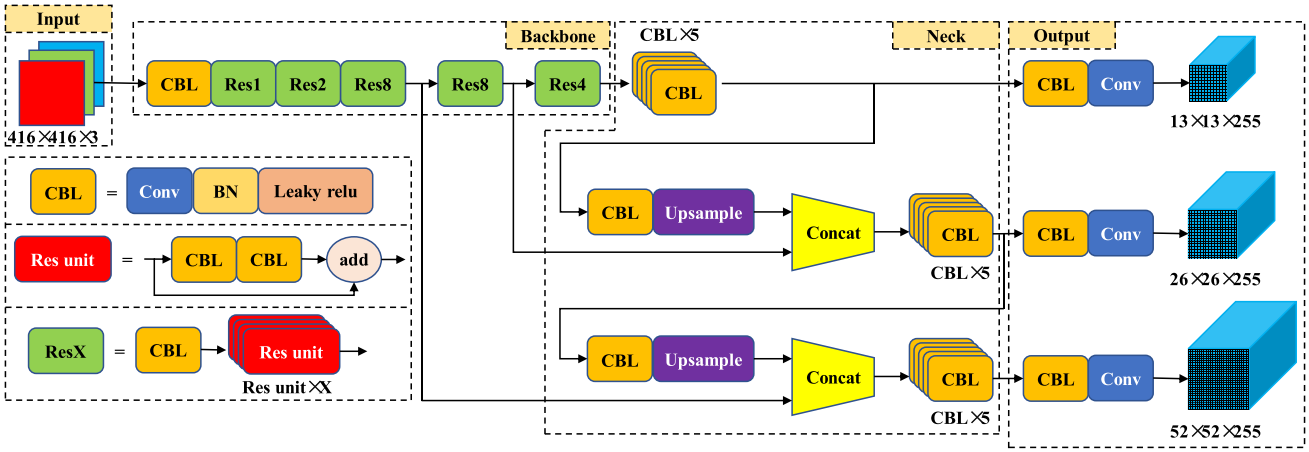

Fig. 4 Original network architecture of YOLOv3

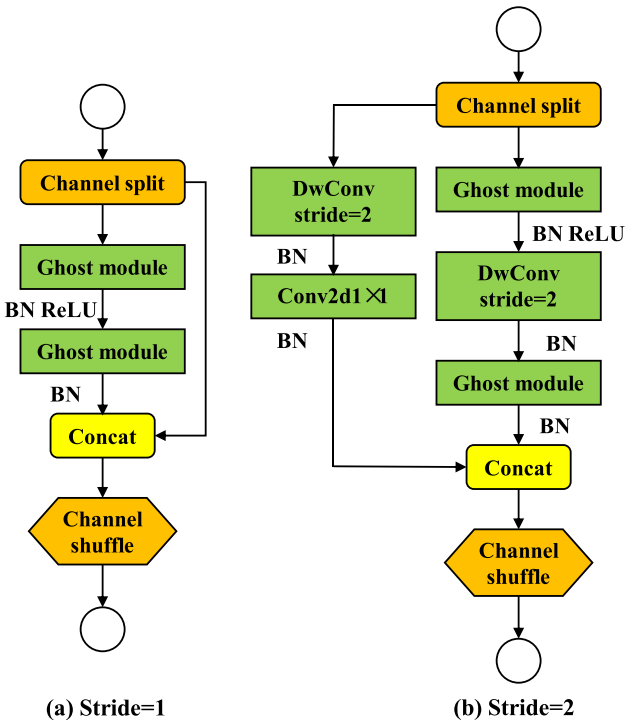

Fig. 5 Ghost-Shuffle bottleneck (GSbneck)

tion is $416^2 \times 3$, the FLOPs of Large and Small models are reduced by 74% and 75% respectively. In terms of model inference speed, Large and Small models can reach 4.3 FPS and 6.1 FPS on CPU (and real-time detection speed of 69 FPS and 87 FPS on GPU), respectively, which are about 4× and 5.5× faster than the original YOLOv3 model. Therefore, the proposed object detection models realize the light-weight design while ensuring the original average accuracy, which lays the foundation for the later visual servoing experiments on the GPU-free bolster spring grasping test platform.

3.4 Training on Custom Dataset

Two thousand and four hundred bolster spring images were collected under different backgrounds and perspectives to construct a bolster spring dataset. LabelImg is used to label all the images in the dataset one by one, and the class ID and normalized coordinate information formed after labeling are used for the training of the proposed MGBM-YOLO models. Some bolster spring images with labels are shown in Fig. 8.

The hardware configuration for training is the same as that in Section 3.3. K-means clustering algorithm is beneficial to speed up the loss reduction during training and improve the detection accuracy of the model [26–28]. Therefore, this paper adopts the K-means clustering algorithm to calculate the anchor boxes of the bolster spring. The clustering results are shown in Table 2. The trained detection model is used for real-time object detection in visual servoing process.

4 IBVS Based on the Corner Points Features of the MGBM-YOLO Object Detection Bounding Box

In this section, an IBVS method based on MGBM-YOLO object detection bounding box corner points features is proposed to realize the visual servoing grasping task of bogie bolster spring. This method assumes that the camera imag-

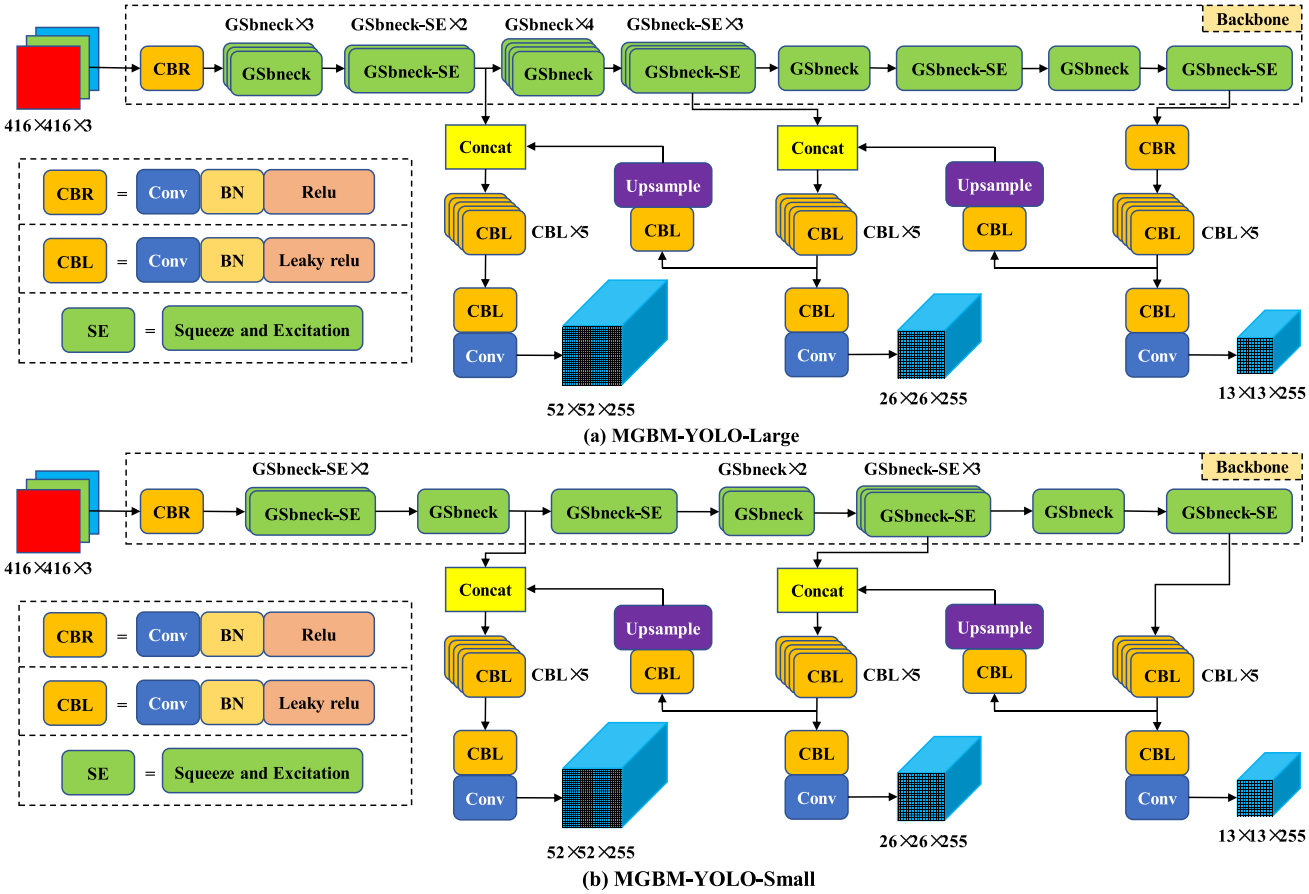

Fig. 6 Network architectures of MGBM-YOLO

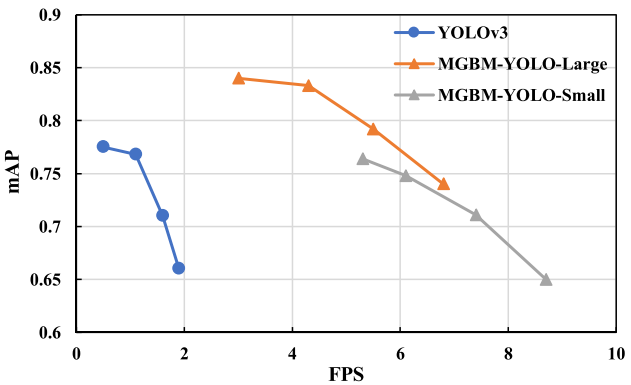

Fig. 7 Accuracy and speed comparison between MGBM-YOLO models and original YOLOv3 model on VOC2007

ing plane is always parallel to the object plane in the initial pose. The detection and positioning of the bolster spring are realized by the object detection algorithm based on CNN, which effectively overcomes the problem that it is difficult to extract the features of the bolster spring image under the complex background by the conventional image processing algorithm.

4.1 Camera-Robot Configuration

The Eye-in-Hand configuration can expand the working range of the camera and improve the image resolution, so the Eye-in-Hand configuration is selected. The Eye-in-Hand system configuration consists of a 6-DOF robot, a camera mounted on the robot end effector, and a dedicated gripper for bolster spring grasping, as shown in Fig. 9. In Fig. 9, T denotes the transformation between the two coordinate frames.

4.2 Robot Kinematics and Jacobian Matrix

4.2.1 Robot Kinematic Model

The ABB IRB4600 robot consists of six revolute joints, among which the axes of joint 4, joint 5 and joint 6 intersect at the same point, forming a so-called spherical wrist. Based on the classic D-H parameter method

Table 1 Comparison of MGBM-YOLO models and original YOLOv3 model

Model	Model size (MB)	Parameters (M)	FLOPs (G)	mAP	FPS/ CPU	FPS/ GPU	Hardware configuration
YOLOv3	236.2	61.626	32.829	0.768	1.1	18	CPU: Intel Core i7 7700
MGBM-YOLO-Large	96.9	23.098	8.537	0.833	4.3	69	
MGBM-YOLO-Small	80.6	21.150	8.152	0.748	6.1	87	GPU: NVIDIA RTX 3070

Note: hardware configuration is for all models

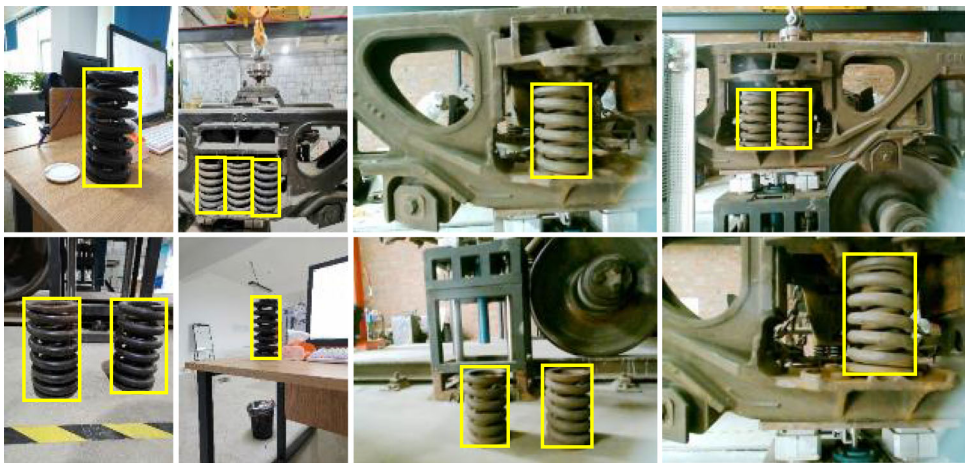

Fig. 8 Custom dataset of bolster spring (labeled)

Table 2 K-means clustering results of anchor boxes

Clustering algorithm	Multi-scale	Anchor boxes
K-means	Scale 1	(104×168), (133×216), (175×297)
	Scale 2	(67×108), (75×118), (88×139)
	Scale 3	(49×78), (55×87), (62×98)

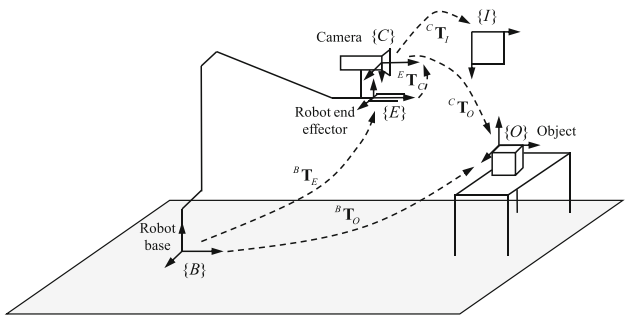

Fig. 9 Robotic Eye-in-Hand system configuration

in the robotics field [29], the kinematic model of the ABB IRB4600/60 kg industrial robot is established. The corresponding parameters of the six joints are listed in Table 3. In Table 3, θ, d, a, and α denote joint angle, link offset, link length, and link twist angle, respectively.

Table 3 ABB IRB4600 D-H parameters

Joint	θ/ °	d/m	a/m	α/°
1	θ_1	0.295	0.175	−90
2	θ_2	0.264	0.9	0
3	θ_3	−0.264	0.175	−90
4	θ_4	0.96	0	90
5	θ_5	0	0	−90
6	θ_6	0.135	0	0

4.2.2 Robot Jacobian Matrix

In the field of robotics, the robot Jacobian matrix reveals the mapping relationship between the velocity of the robot end effector in Cartesian space and the joint velocity, which is defined as follows:

$$ {}^{B}\boldsymbol{v} = \begin{bmatrix} {}^{B}\boldsymbol{v} \\ {}^{B}\boldsymbol{\omega} \end{bmatrix} = {}^{B}\boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{1} $$

where $\dot{\theta}$ and ${}^{B}v$ denote the changing rate of robot joint coordinates (i.e., joint velocity) and the spatial velocity of the end effector in the robot base coordinate frame, respectively. ${}^{B}v$ contains translational and rotational velocity components. ${}^{B}\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbf{R}^{6 \times N}$ is the robot Jacobian matrix (also called geometric Jacobian matrix) in the robot base coordinate frame, and N indicates the number of joints of the robot.

However, in the application of robot visual servoing, in order to establish the mapping relationship between the velocity of the robot end effector in the end effector coordinate frame and the joint velocity, it is usually imperative to obtain the robot Jacobian matrix in the end effector coordinate frame shown in Eq. (2):

$$^{E}\boldsymbol{v}=\begin{bmatrix}{}^{E}\boldsymbol{v}\\ {}^{E}\boldsymbol{\omega}\end{bmatrix}={}^{E}\boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{2}$$

The robot Jacobian matrix in the end effector coordinate frame can be obtained by multiplying the robot Jacobian matrix in the base coordinate frame by the transformation matrix between the two coordinate frames, such that

$$^{E}\boldsymbol{J}(\boldsymbol{\theta})=\begin{bmatrix}{}^{E}_{\mathrm{B}}\boldsymbol{R} & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & {}^{E}_{\mathrm{B}}\boldsymbol{R}\end{bmatrix}{}^{\mathrm{B}}\boldsymbol{J}(\boldsymbol{\theta}) \tag{3}$$

where ${}^{E}_{\mathrm{B}}\boldsymbol{R}$ is the transpose of the rotation matrix between the robot base coordinate frame and the end effector coordinate frame, which can be calculated by forward kinematics from the D-H parameters in Section 4.2.1.

4.3 Selection of Image Features and Depth-Independent Image Jacobian Matrix

In traditional vision systems, feature points are widely used as image features because of their simple extraction and easy derivation of the relationship between feature changes and camera pose changes. However, for bolster spring images with complex geometric features in a cluttered background, conventional image processing algorithms are difficult to extract the desired points features. MGBM-YOLO object detection algorithm based on deep learning proposed in Section 3 can effectively detect and position the bolster spring, and output accurate bounding box, which provides a good idea for feature selection of bolster spring image.

As shown in Fig. 10, the yellow bounding box is the detecting and positioning result of the bolster spring obtained by the object detection algorithm. The pixel coordinates of the four corners (shown in red) of the bolster spring bounding box on the image plane are expressed as $c_1 = (u_1, v_1), c_2 = (u_2, v_2), c_3 = (u_3, v_3), c_4 = (u_4, v_4)$, respectively, and these four coordinates are selected as the image features of the feature points-based visual servoing.

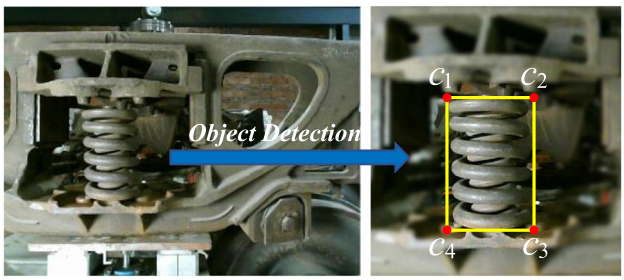

Fig. 10 Selection of image features

In the IBVS system, similar to the robot Jacobian matrix, the image Jacobian matrix is used to establish the mapping relationship between the image feature space and the robot end motion space, and its expression is as follows:

$$\dot{\boldsymbol{y}}=\boldsymbol{J}(\boldsymbol{r})\dot{\boldsymbol{r}} \tag{4}$$

where $\dot{\boldsymbol{r}}\in\mathbf{R}^{6}$ denotes the changing rate of the pose of the robot end effector in end effector coordinate frame (i.e., ${}^{E}\boldsymbol{v}$ in Eq. (2)). $\dot{\boldsymbol{y}}\in\mathbf{R}^{m}$ is the changing rate of image features, and its dimension m is the number of image features. $\boldsymbol{J}(\boldsymbol{r})\in\mathbf{R}^{m\times6}$ stands for the image Jacobian matrix.

For the specific 6-DOF robot visual servoing system and the selected corner points features of the bounding box, the specific analytical expression of image Jacobian matrix has been derived in [2], such that

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix}=\begin{bmatrix}-\frac{f}{Z} & 0 & \frac{x}{Z} & \frac{xy}{f} & -\frac{f^2+x^2}{f} & y\\ 0 & -\frac{f}{Z} & \frac{y}{Z} & \frac{f^2+y^2}{f} & -\frac{xy}{f} & -x\end{bmatrix}\begin{bmatrix}\nu_x\\ \nu_y\\ \nu_z\\ \omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \tag{5}$$

where $\boldsymbol{f}$ denotes the focal length of the camera, (x, y) denotes the image coordinate, and Z stands for the depth value of the feature point in the camera coordinate frame. According to the relationship between the image plane coordinate frame and the pixel coordinate frame, the relationship between the changing rate of image features in pixel coordinate form and the camera velocity can be obtained, see Eq. (6), where $\boldsymbol{\rho_w}$ and $\boldsymbol{\rho_h}$ are the pixel sizes, respectively representing the width and height of each pixel. $\boldsymbol{u_0}$ and $\boldsymbol{v_0}$ denote the abscissa and ordinate of the principal point respectively.

$$\begin{bmatrix}\dot{u}\\ \dot{v}\end{bmatrix}=\begin{bmatrix}-\frac{f}{\rho_w Z} & 0 & \frac{(u-u_o)}{Z} & \frac{\rho_w(u-u_o)(v-v_o)}{f} & -\frac{f^2+\rho_w^2(u-u_o)^2}{\rho_w f} & \frac{\rho_h}{\rho_w}(v-v_o)\\ 0 & -\frac{f}{\rho_h Z} & \frac{v-v_o}{Z} & \frac{f^2+\rho_h^2(v-v_o)^2}{\rho_h f} & -\frac{\rho_h(u-u_o)(v-v_o)}{f} & -\frac{\rho_w}{\rho_h}(u-u_o)\end{bmatrix}\begin{bmatrix}v_x\\ v_y\\ v_z\\ \omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \tag{6}$$

The relationship in Eq. (6) can be directly applied to the corner points features of the bounding box used in this paper. Furthermore, an online feature depth estimation method based on image area feature is proposed, which assumes that the camera projection geometry is modeled by perspective projection. To simplify the mathematical derivation, the pinhole camera model as shown in Fig. 11 is adopted.

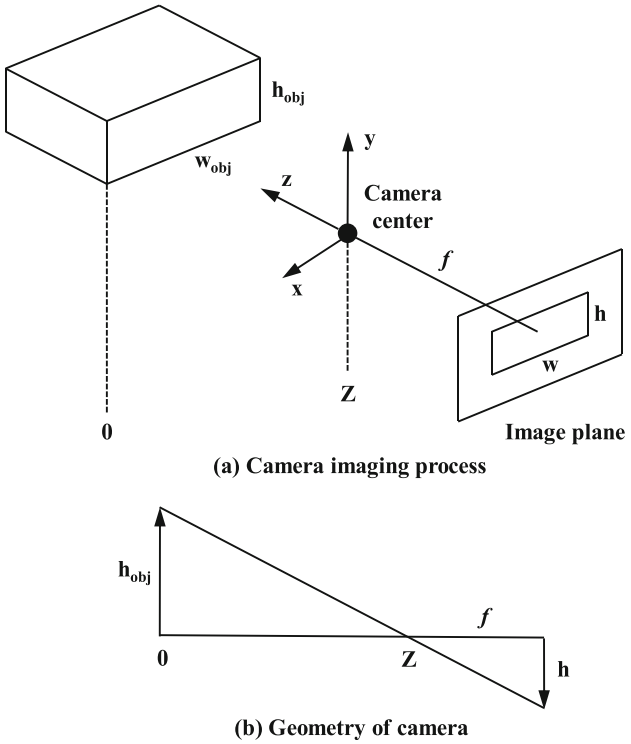

Fig. 11 Pinhole camera model

Figure 11a shows the imaging process of the pinhole camera. Z is assumed to be the estimated depth of the object with respect to the camera, and f denotes the focal length of the camera. The geometric imaging relationship of the pinhole camera observed from the x-axis can be seen in Fig. 11b, which clearly shows the projection relationship in the height direction of the object. Similarly, the projection relationship in the width direction of the object can be observed from the y-axis. By analyzing the above projection geometry, the following relationships can be obtained:

$$\begin{cases} h = \frac{f}{Z} \times h_{obj} \\ w = \frac{f}{Z} \times w_{obj} \end{cases} \tag{7}$$

where h_{obj} and w_{obj} denote the projection height and width of the object in the image plane respectively, and h and w denote the actual height and width of the object respectively. Combining $A = w \times h$, the relationship between the image area and the object depth can be obtained as

$$\frac{A}{A^*} = \frac{Z^{*2}}{Z^2} \tag{8}$$

where A and A^* denote the image area calculated in real time and the desired area, respectively. Z and Z^* denote the feature depth estimated in real time and the desired feature depth. It should be noted that, since the object plane and the image plane are always parallel in the visual grasping application of the bolster spring, it is very easy to calculate the image area of the bolster spring in real time according to the width and height of the MGBM-YOLO object detection bounding box. At the same time, the premise that the object is parallel to the image plane also ensures that the online depth estimation based on the aforementioned area feature of bolster spring is reasonable.

Finally, based on the above discussion and in the light of Eq. (8), the image Jacobian matrix in Eq. (6) can be rewritten as shown in Eq. (9).

It can be seen from Eq. (9) that the defined $J(r)$ becomes the depth-independent image Jacobian matrix due to the substitution of $\sqrt{A}$. Compared with the feature depth Z, the area feature $\sqrt{A}$ of bolster spring can be obtained more easily and accurately when only monocular camera is adopted.

4.4 IBVS Controller Design

The industrial robot adopted in this paper is ABB IRB4600. When the joint receives the external control signal, the controller system usually receives the velocity of the joint. Therefore, it is imperative to establish the differential mapping relationship between the image feature space and the robot joint space before designing the visual servoing controller. Based on the robot Jacobian matrix and the image Jacobian matrix proposed in Sections 4.2 and 4.3 respectively, the above differential mapping relationship can be obtained by substituting Eq. (2) into Eq. (4):

$$\boldsymbol{J}(\boldsymbol{r}) = \begin{bmatrix} -\frac{f\sqrt{A}}{\rho_w Z^*\sqrt{A^*}} & 0 & \frac{(u-u_o)\sqrt{A}}{Z^*\sqrt{A^*}} & \frac{\rho_w(u-u_o)(v-v_o)}{f} & -\frac{f^2+\rho_w{}^2(u-u_o)^2}{\rho_w f} & \frac{\rho_h}{\rho_w}(v-v_o) \\ 0 & -\frac{f\sqrt{A}}{\rho_h Z^*\sqrt{A^*}} & \frac{(v-v_o)\sqrt{A}}{Z^*\sqrt{A^*}} & \frac{f^2+\rho_h{}^2(v-v_o)^2}{\rho_h f} & -\frac{\rho_h(u-u_o)(v-v_o)}{f} & -\frac{\rho_w}{\rho_h}(u-u_o) \end{bmatrix} \tag{9}$$

$$\dot{\boldsymbol{y}} = \boldsymbol{J}(\boldsymbol{r})\dot{\boldsymbol{r}} = \boldsymbol{J}(\boldsymbol{r})\boldsymbol{J}(\boldsymbol{\theta})\dot{\theta} \tag{10}$$

The visual servoing controller is designed to minimize the feature error, which is usually defined as:

$$\boldsymbol{e}(\boldsymbol{y}) = \boldsymbol{y} - \boldsymbol{y}^* \tag{11}$$

where $\boldsymbol{y}$ and $\boldsymbol{y}^*$ denote the current image feature and the desired image feature, respectively.

Proportional controller is widely used in the domain of robot visual servoing because of its simplicity and maturity. $\dot{e}(y) = -\lambda e(y)$ is employed to make the feature error $e(y)$ decrease exponentially, and then a simple proportional controller can be obtained in light of Eqs. (10) and (11):

$$\dot{\boldsymbol{\theta}} = -\boldsymbol{\lambda} J(\boldsymbol{\theta})^{\dagger} \boldsymbol{J}(\boldsymbol{r})^{\dagger} \boldsymbol{e}(\boldsymbol{y}) \tag{12}$$

where $\boldsymbol{\lambda}$ is the proportional gain of the controller, $\boldsymbol{J}(\boldsymbol{\theta})^{\dagger}$ and

$\boldsymbol{J}(\boldsymbol{r})^{\dagger}$ denote the pseudo-inverse of the robot Jacobian and the image Jacobian, respectively. $\dot{\boldsymbol{\theta}}$ indicates the joint velocity input to the robot joint controller.

Considering that the observed object is easy to exceed its field of view when it is close to the webcam, the classic control strategy of approach-align- grasp is adopted in the bolster spring grasping system. In the approaching phase, the dedicated gripper for the bolster spring installed at the robot end is driven to the desired pose by the IBVS method based on the corner points features of MGBM-YOLO object detection bounding box. Then the dedicated gripper moves the fixed distance measured in the teaching pose to achieve the alignment of the bolster spring. Finally, the dedicated gripper is employed to achieve the grasping task of the bolster spring. The norm of the image feature error is used as the condition for judging the end of the visual servoing iterations, so Eq. (12) can be rewritten as

$$\begin{cases} \dot{\boldsymbol{\theta}} = -\lambda \boldsymbol{J}(\boldsymbol{\theta})^{\dagger} \boldsymbol{J}(\boldsymbol{r})^{\dagger} \boldsymbol{e}(\boldsymbol{y}), & \|\boldsymbol{e}(\boldsymbol{y})\| \geq e_0 \\ \text{Stop servoing}-\text{Start aligning\&grasping}, & \text{otherwise} \end{cases} \tag{13}$$

where $\boldsymbol{e_0}$ is a predetermined threshold value. When the error norm decreases to within the threshold range, the visual servoing approaching phase ends, the specific aligning and grasping actions of robot are performed.

The block diagram of the IBVS control system based on the corner points features of the object detection bounding box is shown in Fig. 12. The webcam mounted on the robot end effector captures object images at a specific sampling rate, and then the MGBM-YOLO object detection algorithm is employed to achieve the detecting and positioning of the bolster spring. The corner points coordinates of the bolster spring bounding box are selected as the image features, and the IBVS controller is designed based on the deviation between the current image features captured by the webcam and the desired image features. It should be noted that the proposed feature depth online estimation method is employed in the IBVS controller to obtain accurate depth-independent image Jacobian matrix. The joint angles calculated by the IBVS controller in real time are transmitted to the robot joint controller to drive the robot joints to the desired pose, thereby forming an IBVS closed-loop control system.

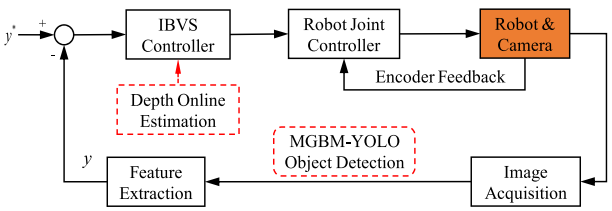

Fig. 12 Block diagram of the IBVS control system based on the corner points features of the object detection bounding box

5 Experimental Validation

In this section, to validate the effectiveness and stability of the modified MGBM-YOLO-Large and MGBM-YOLO-Small object detection models and the proposed depth-independent image Jacobian matrix for the bolster spring visual servoing grasping system under cluttered background, two groups of comparative experiments have been conducted on the robotic grasping physical platform of bolster spring as shown in Fig. 13, including the visual servoing grasping experiments based on the MGBM-YOLO-Large and MGBM-YOLO-Small models under the online estimated depth and the desired depth, respectively.

5.1 Experimental Setup

It can be seen from Fig. 13 that the robotic grasping physical platform of bolster spring is composed of an industrial robot ABB IRB4600 with IRC5 controller, a dedicated gripper for bolster spring installed on the end flange of the robot, a webcam installed on the gripper (the optical axis of the webcam is parallel to the 6th-axis of the robot) and a NORCO industrial PC. One of the advantages of the ABB robot's IRC5 controller is that it can receive the joint angle and drive the robot to the corresponding position through the command MoveAbsJ, which is very convenient for receiving the control output of the designed visual servoing controller. However, the weakness of the IRC5 controller is that it cannot achieve real-time control without activating the External Guided Motion (EGM) control module. More precisely, when the IBVS controller generates a new set of joint coordinates and sends it to the IRC5 controller, it will not respond until the previous joint angle is reached in each iteration. Due to the limitation of this hardware condition, the response time of the robot controller can be balanced by setting a smaller proportional gain and larger camera sampling time. In the two comparative experiments, the proportional gain of the IBVS controller and the camera sampling time are set as $\boldsymbol{\lambda} = 0.02$ and 0.5 s, respectively.

Based on the advantages of high accuracy and simple operation [30], Zhang's Calibration Method is adopted to obtain the intrinsic parameters of the webcam. Firstly, 15 checkerboard images were captured from different perspectives and distances. Then, the Camera Calibrator in MATLAB is used to calibrate the webcam according to the images of checkerboard. The checkerboard corners detection results are shown in Fig. 14, and the calibration results of the webcam intrinsic parameters are listed in Table 4.

The camera intrinsic parameters involved in the image Jacobian matrix employ the calibration results in Table 4. In other words, the focal length, resolution and principal point coordinate of the webcam are set as 0.0015 m, 640 × 480

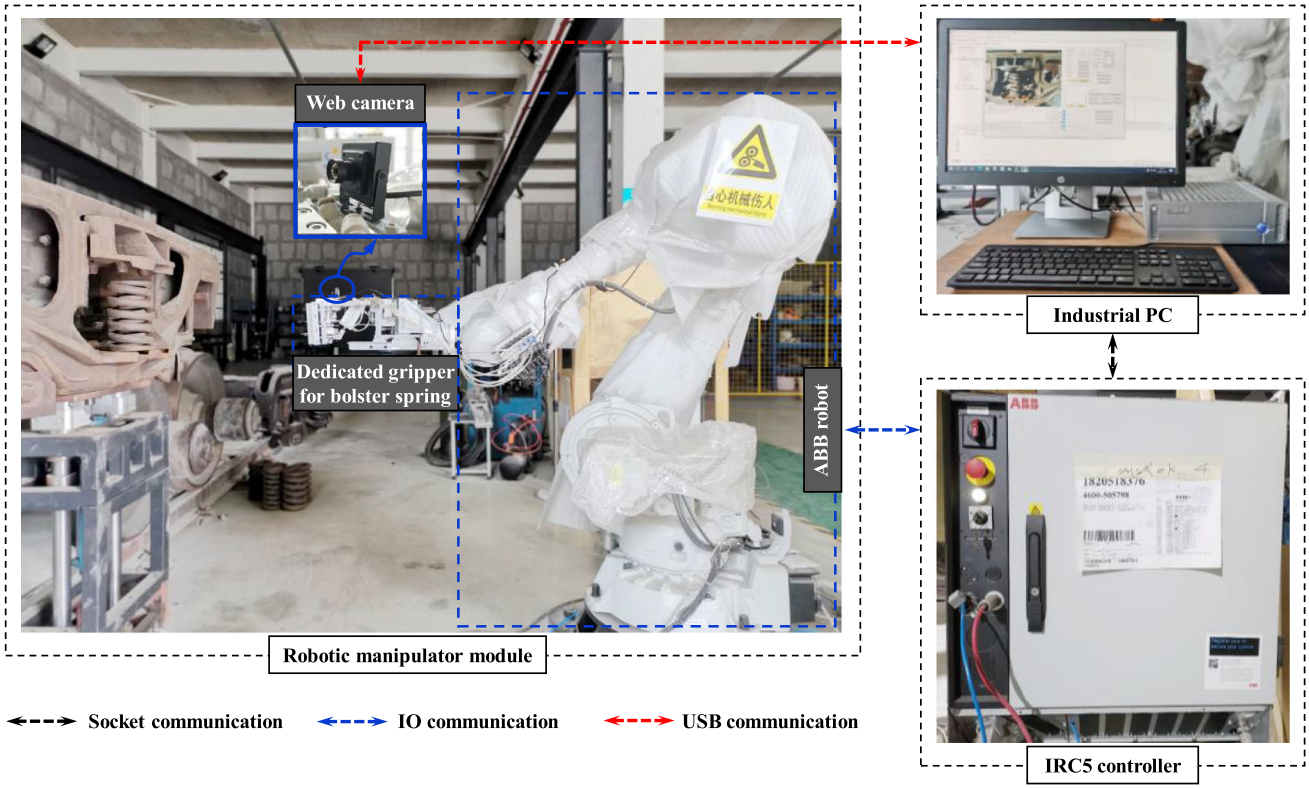

Fig. 13 Robotic grasping physical platform of bolster spring

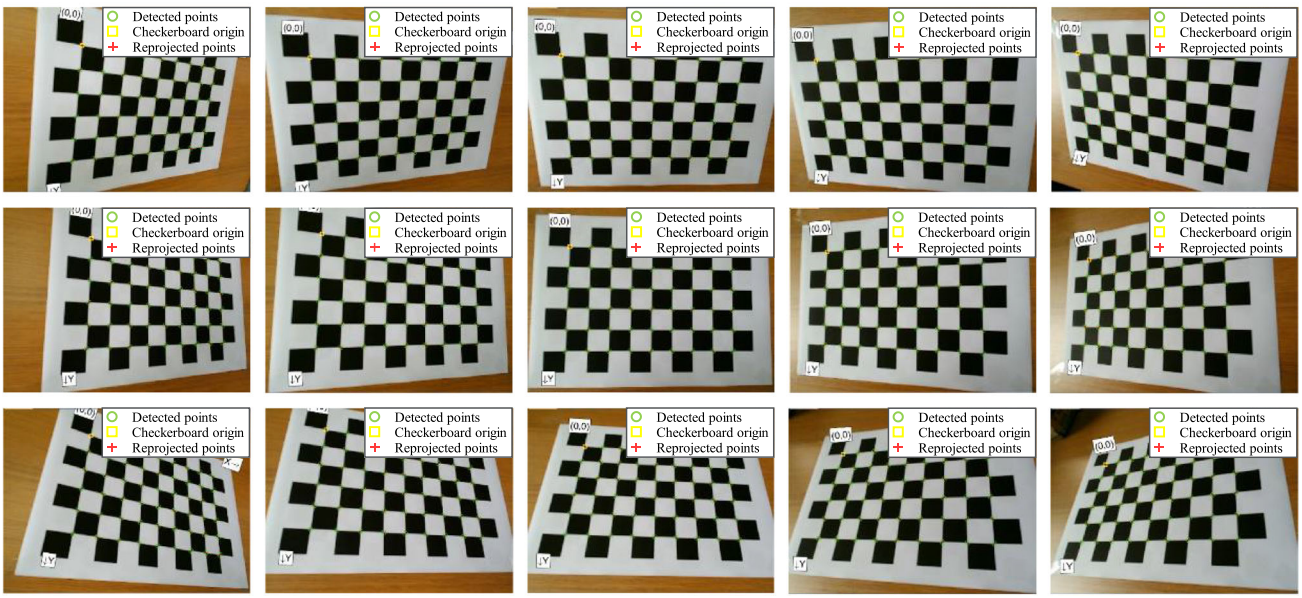

Fig. 14 Corners detection results of checkerboard

and (335.183, 230.758), respectively. According to the specifications of the webcam, the pixel size $\rho_w = \rho_h = 3\mu m$. The desired image feature obtained by teaching is $y^* = [241, 377, 377, 241, 104, 104, 327, 327]$ and the desired depth $Z^* = 0.4m$. The threshold value $\boldsymbol{e_0}$ in Eq. (18) is set as 20 ***pixels***, which corresponds to an actual size of approximately 15 mm.

Table 4 Calibration results of the webcam intrinsic parameters

Method	Number of patterns	Image size	Focal length/m	Principal point
Zhang's Calibration Method	15	640× 480	0.0015	(335.183, 230.758)

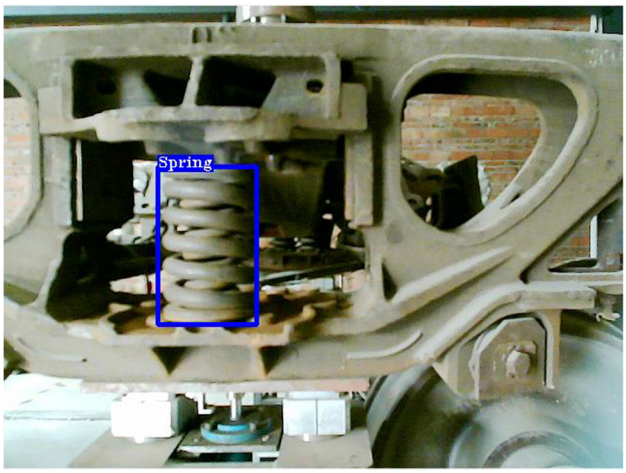

Fig. 15 Image of bolster spring in initial pose

5.2 Results Analysis

To ensure that the imaging plane of the webcam is parallel to the object plane in the initial pose of the robot, the initial joint coordinate θ_0 of the robot is set as [−29.72°, −35.55°, 70.07°, −43.03°, −44.18°, 34.66°], and the corresponding initial pose image is shown in Fig. 15. It is worth pointing out that, except for the different control variables, the parameters of the two comparative experiments are all set to the same. Repeated experiments are conducted to validate the effectiveness and reliability of visual servoing. The experimental results show good consistency in both the iterations and the final poses of system convergence. Limited to the length of the paper, the experimental data such as image features trajectories, feature errors and camera velocity in the randomly selected two groups of comparison tests are displayed, as shown in Figs. 16 and 17.

The left and right columns of Fig. 16 respectively show the experimental results of visual servoing based on the MGBM-YOLO-Large object detection bounding box corner points features using the proposed online estimated depth and the desired depth. It can be seen from Fig. 16a and b that the image features under the two different depth estimation methods both converge from the initial position to the desired position. For the feature errors, when the desired feature depth value is adopted, the visual servoing system needs more than 60 iterations to converge within the predetermined threshold range (see Fig. 16d). However, when the online depth estimation method based on the area feature of the bolster spring bounding box proposed in this paper is employed, the system convergence process only iterates about 50 times (see Fig. 16c). The camera velocity curves in Fig. 16e and f also converge to zero in the corresponding iteration period. The above experimental results show that the proposed MGBM-YOLO-Large object detection model can stably detect and position the bolster spring under the cluttered background of the bogie. The adoption of the proposed depth-independent image Jacobian matrix calculated with the online estimated depth in the visual servoing system can effectively accelerate the system convergence speed.

From the comparison results of the detection performance on the Pascal VOC dataset in Section 3.3, it can be seen that the MGBM-YOLO-Small model has a faster detection speed than the MGBM-YOLO-Large model, but the accuracy is slightly poor. In theory, if the positioning accuracy of the Small model can meet the requirements of the bolster spring visual servoing system, the high-accuracy Large model can also meet the requirements. Therefore, to validate the effectiveness of the MGBM-YOLO-Small model for bolster spring visual servoing grasping, a more complete experiment was carried out using the approach-align-grasp control strategy. The experimental results of the visual servoing approaching phase are shown in Fig. 17. The results show that the IBVS controller designed by using the Small model to detect the corner points features of the bounding box can also successfully converge the image features to the desired position. At the same time, using the feature depth online estimation method proposed in this paper has a significant effect on improving the system convergence speed. Convergence of the image features to the desired position means that the end effector of the bolster spring grasping robot in Cartesian space has also moved to the desired pose. Relative to the pre-captured position of the desired image feature, the residual errors of the robot's translational and rotational motions are less than 5 mm and 1°, respectively. Because the dedicated gripper used for bolster spring has a certain degree of flexibility and adaptability, this positioning accuracy level is sufficient for the grasping of bolster spring.

Figure 18 shows the successful grasping process of the bolster spring after the visual servoing in the cluttered background of the bogie. At the end of the visual servoing approaching phase, the pose reached by the dedicated gripper for bolster spring is shown in Fig. 18a. Then the dedicated gripper for the bolster spring follows the robot to move 0.4 m along the optical axis of the camera (the

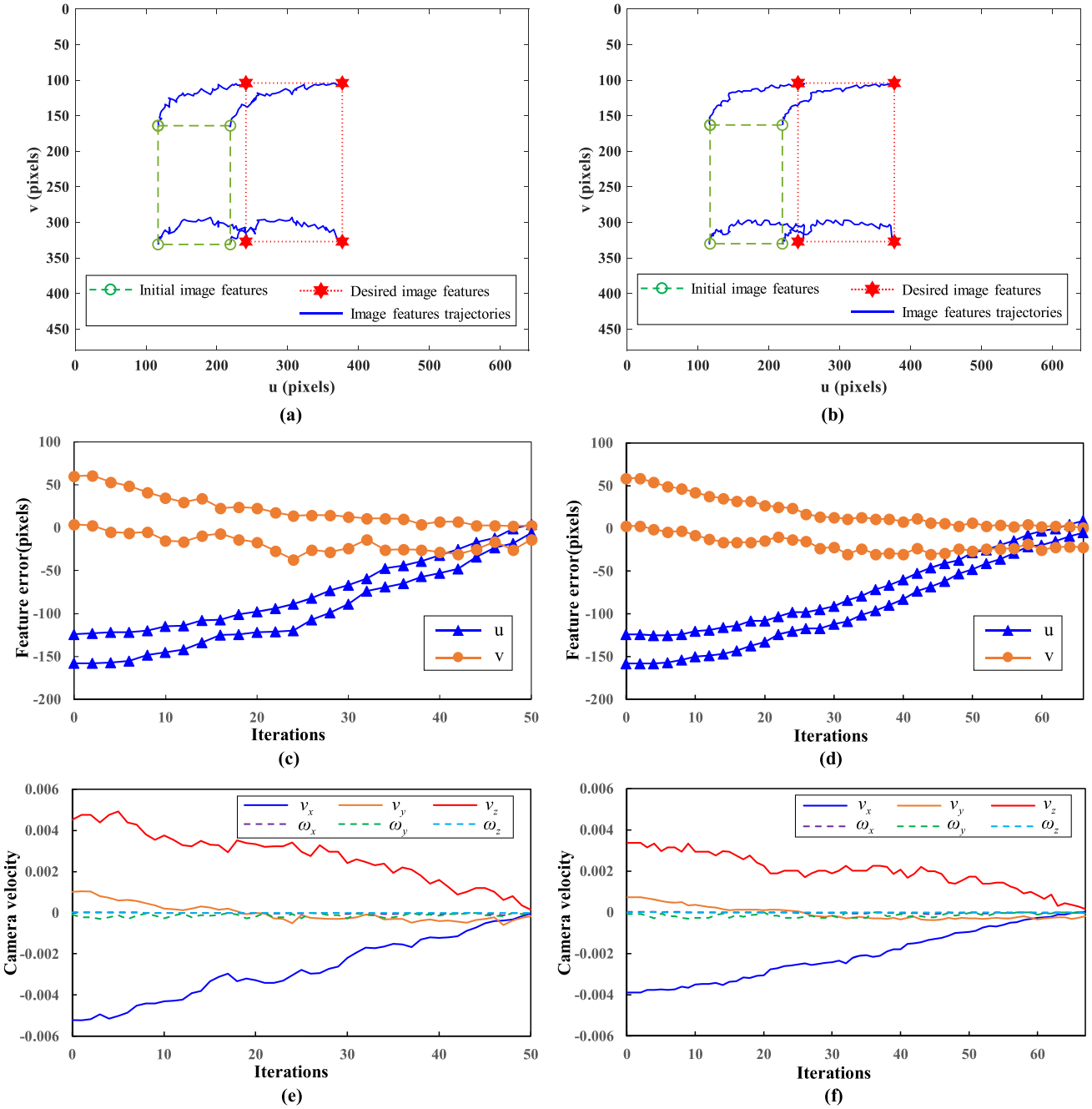

Fig. 16 Experimental results of visual servoing based on MGBM-YOLO-Large model (Left: using online estimated depth. Right: using desired depth.): **a**, **b** image features trajectories, **c**, **d** feature errors curve, **e**, **f** camera velocity curve (m/s, rad/s)

depth value measured in the desired pose) to realize the alignment of the gripper and the bolster spring, as shown in Fig. 18b. Finally, the dedicated gripper successfully accomplishes the grasping of the bolster spring, as shown in Fig. 18c. The experimental results show that the proposed MGBM-YOLO-Small object detection model can accurately and stably realize the visual servoing grasping of bolster spring with complex geometric features under cluttered backgrounds.

6 Conclusion and Future Work

Based on the MobileNetv3 network architecture and GhostNet bottleneck layer module, this paper proposes two modified light-weight YOLO models: MGBM-YOLO-Large and MGBM-YOLO-Small. Compared with the original YOLOv3 model, the sizes of the MGBM-YOLO-Large and MGBM-YOLO-Small models are reduced by 59% and 66%, respectively, and the model parameters

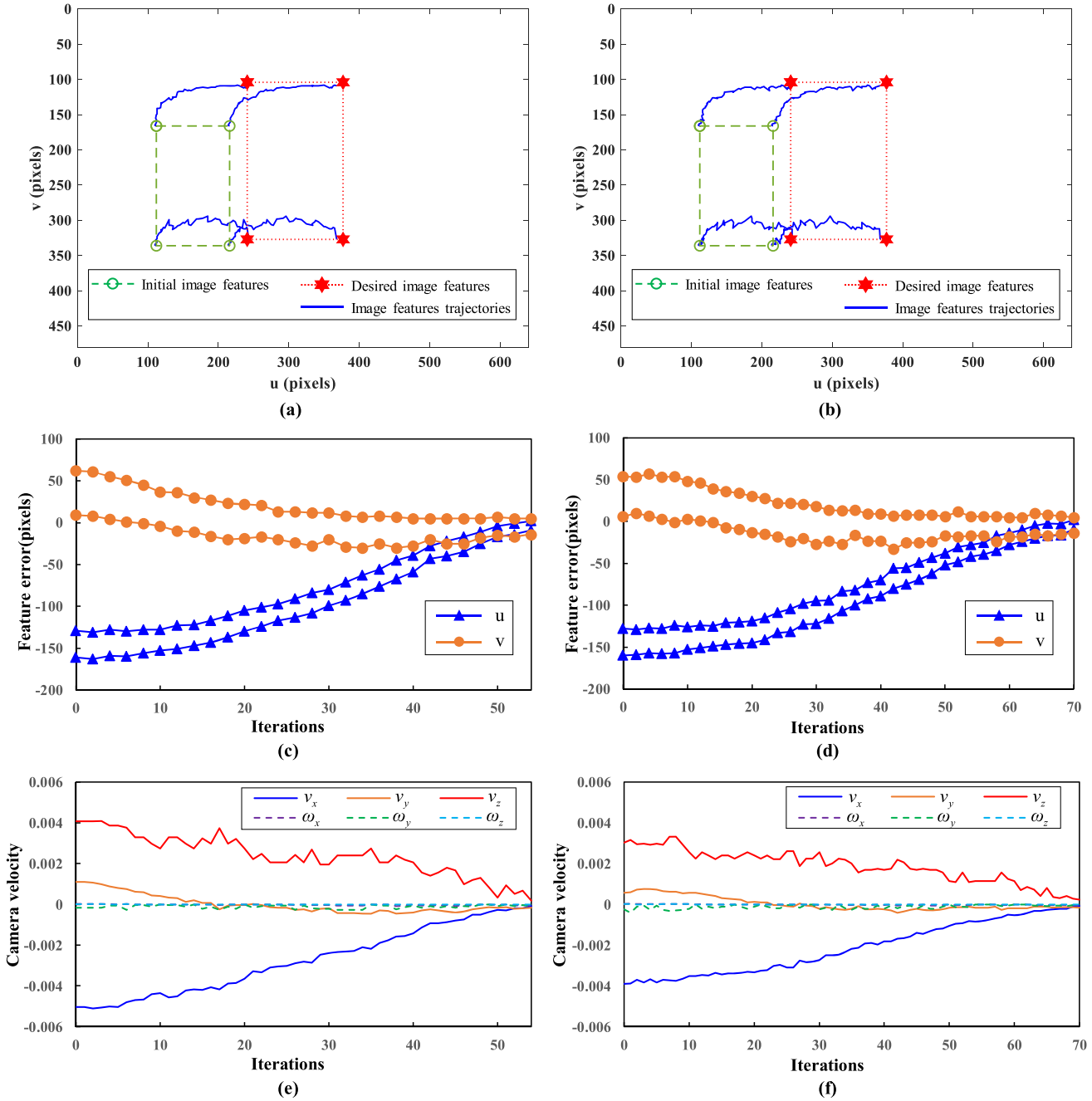

Fig. 17 Experimental results of visual servoing based on MGBM-YOLO-Small model (Left: using online estimated depth. Right: using desired depth.): **a**, **b** image features trajectories, **c**, **d** feature errors curve, **e**, **f** camera velocity curve (m/s, rad/s)

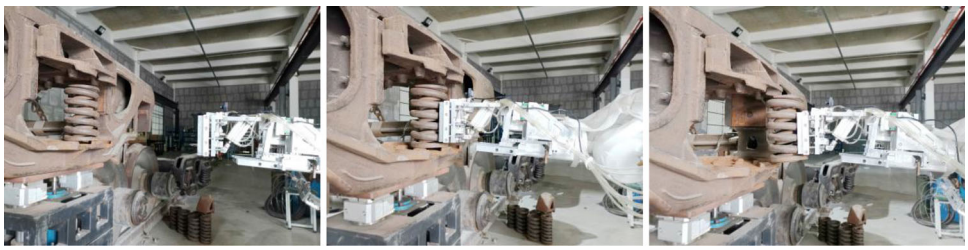

Fig. 18 Aligning-grasping process of bolster spring based on MGBM-YOLO-Small model

are reduced by 63% and 66%, respectively. The training and evaluation results on the Pascal VOC2007 and Pascal VOC2012 datasets show that the detection speeds of the proposed MGBM-YOLO-Large and MGBM-YOLO-Small models are 4× and 5.5× faster respectively than the original YOLOv3 model while maintaining the comparable average accuracy. Based on the MGBM-YOLO models, an IBVS method based on the corner points features of the object detection bounding box is proposed to grasp the bolster spring with complex geometric features in a cluttered background. To accelerate the convergence speed of the IBVS system, this paper proposes an online feature depth estimation method that associates the depth information with the area feature of the detection bounding box of the bolster spring and applies it to the IBVS system based on the corner points features of MGBM-YOLO object detection bounding box. Two groups of comparative experiments with different feature depths and different object detection models were conducted on the robotic grasping physical platform of bolster spring. The experimental results show that the two MGBM-YOLO detection models proposed in this paper can meet the real-time detecting and positioning requirements of the IBVS-based robot grasping system of bolster spring, and the proposed feature depth online estimation method is significant for accelerating the convergence speed of the visual servoing system.

Future work includes introducing the object detection sloping bounding box to address the problem of visual servoing grasping when the bolster spring and the camera are sloped, and validating the effectiveness of the proposed method on a robot system supporting real-time control.

Code Availability Not applicable.

Authors' Contributions Huanlong Liu: Conceptualization, Methodology. Dafa Li: Software, Experimental validation, Data curation, Writing-Reviewing and Editing. Bin Jiang: Data curation, Visualization. Jianyi Zhou: Investigation. Tao Wei: Investigation. Xinliang Yao: Investigation.

Data Availability Not applicable.

Declarations

Conflict of Interest No conflict of interest exits in the submission of this article, and the article is approved by all authors for publication.

Ethics Approval Not applicable.

Consent to Participate Not applicable.

Consent for Publication Not applicable.

References

1. Xin, J.S., Shang, Y.J., Xue, H., et al.: Dynamic reliability sensitivity analysis of bolster spring for heavy-haul freight Car. J. Lanzhou Jiaotong Univ. **39**(06), 86–91 (2020)
2. Chaumette, F., Hutchinson, S.: Visual servo control, part I: basic approaches[J]. IEEE Robot. Autom. Mag. **13**(4), 82–90 (2006)
3. Malis, E., Ezio, S., Chaumette, F., et al.: 2-1/2-D visual Servoing[J]. IEEE Trans Rob Autom. (1999)
4. Espiau B.: Effect of Camera Calibration Errors on Visual Servoing in Robotics[C]// Preprints of the Third International Symposium on Experimental Robotics. (1993)
5. Li., D.F., Liu, H.L., Wei, T., et al.: Robotic grasping method of bolster spring based on image-based visual servoing with YOLOv3 object detection algorithm[J]. Proc. Inst. Mech. Eng. Part C-J. Mech. Eng. Sci. (2021)
6. Howard, A., Sandler, M., Chu, G., et al.: Searching for Mobilenetv3[C]// 2019 IEEE/CVF international conference on computer vision (ICCV). IEEE, 1314–1324 (2020)
7. Han, K., Wang, Y., Tian, Q., et al.: GhostNet: More Features from Cheap Operations[C]// 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, 1580–1589 (2020)
8. Lin, M., Chen, Q., Yan, S.: Network in network[J]. Comput. Therm. Sci. (2013)
9. Szegedy, C., Liu, W., Jia, Y., et al.: Going Deeper with Convolutions[C]// IEEE Computer Society (2014)
10. Iandola, F. N, Han, S., Moskewicz, M. W., et al.: SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and < 0.5 MB model size[J]. (2016)
11. Howard, A. G., Zhu, M., Chen, B., et al.: Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. (2017)
12. Sandler, M., Howard, A., Zhu, M., et al.: Mobilenetv2: inverted residuals and linear bottlenecks[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520 (2018)
13. Zhang, X., Zhou, X., Lin, M., et al.: ShuffleNet: an extremely efficient convolutional neural network for Mobile devices[C]// 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 6848–6856 (2018)
14. Ma, N., Zhang, X., Zheng, H. T, et al.: ShuffleNet v2: Practical Guidelines for Efficient CNN Architecture Design[C]// European Conference on Computer Vision (ECCV). Springer, Cham, 116–131 (2018)
15. Xu, D.: A tutorial for monocular visual servoing. Acta Aotumatica Sinica. **44**(10), 1729–1746 (2018)
16. Yang, Y., Song, Y., Pan, H., et al.: Visual servo simulation of EAST articulated maintenance arm robot[J]. Fusion Eng. Des. **104**(Mar.), 28–33 (2016)
17. Tongloy, T., Boonsang, S.: An Image-Based Visual Servo Control System Based on an Eye-in-Hand Monocular Camera for Autonomous Robotic Grasping[C]// International Conference on Instrumentation. IEEE (2016)
18. Dong, G.Q., Zhu, Z.H.: Kinematics-based incremental visual servo for robotic capture of non-cooperative target[J]. Robot. Auton. Syst. **112**, 221–228 (2019)

19. Xie, W.F., Li, Z., Tu, X.W., et al.: Switching control of image-based visual Servoing with laser pointer in robotic manufacturing systems[J]. IEEE Trans. Ind. Electron. **56**(2), 520–529 (2009)
20. Mehta, S.S., Ton, C., Rysz, M., et al.: New approach to visual servo control using terminal constraints[J]. J. Frankl. Inst. **356**(10), 5001–5026 (2019)
21. Zhong, X.G., Zhong, X.Y., Hu, H.S., et al.: Adaptive neuro-filtering based visual servo control of a robotic manipulator[J]. IEEE Access. **7**, 76891–76901 (2019)
22. Yang, Y., Zhang, W.S., He, Z.W., et al.: High-speed rail pole number recognition through deep representation and temporal redundancy[J]. Neurocomputing. **415**, 201–214 (2020)
23. Álvaro, A.G., Álvarez-García, J.A., Soria-Morillo, L.M.: Evaluation of deep neural networks for traffic sign detection system[J]. Neurocomputing. **316**, 332–344 (2018)
24. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 779–788 (2016)
25. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 7263–7271 (2017)
26. Redmond, J., Farhadi, A.: YOLOv3: An incremental improvement[J]. arXiv preprint arVix: 1804.02767 (2018)
27. Liu, J., Wang, X.W.: Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network[J]. Front. Plant Sci. **11**, 898 (2020)
28. Xu, D.Q., Wu, Y.Q.: MRFF-YOLO: a multi-receptive fields fusion network for remote sensing target detection[J]. Remote Sens. **12**, 3118 (2020)
29. Corke, P.: Robotics, Vision and Control[M]// Robotics, Vision and Control - Fundamental Algorithms in MATLAB®. Springer, Berlin (2011)
30. Zhang, Z.Y.: A flexible new technique for camera calibration[J]. IEEE Trans. Pattern Anal. Mach. Intell. (2000)

Huanlong Liu received the B.S. M.S. and Ph.D. degrees from the School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, China, in 1999, 2002 and 2008, respectively. He is currently an associate professor in the School of Mechanical Engineering, Southwest Jiaotong University. His research interests include industrial robot technology and intelligent equipment, electromechanical hydraulic hybrid technology and fluid power and cleaning technology.

Dafa Li received the B.S. degree in Mechanical Engineering from Southwest Petroleum University, Chengdu, China, in 2019. He is currently a postgraduate in the School of Mechanical Engineering, Southwest Jiaotong University. His research interests include robotics, machine vision, deep learning and visual servoing.

Bin Jiang received the B.S. degree in Aircraft Manufacture Engineering from North China Institute of Aerospace Engineering, Langfang, China, in 2016. He is currently a postgraduate in the Graduate School of Tangshan, Southwest Jiaotong University. His research interests include industrial intelligent equipment and machine vision.

Jianyi Zhou received the B.S. degree in Mechanical Design and Manufacturing and Its Automation from Southwest Jiaotong University, Chengdu, China, in 2019. He is currently a postgraduate in the School of Mechanical Engineering, Southwest Jiaotong University. His research interests include industrial intelligent equipment and electromechanical hydraulic hybrid technology.

Tao Wei received the B.S. degree in Mechanical Engineering from Zhejiang University of Technology, Hangzhou, China, in 2020. He is currently a postgraduate in the School of Mechanical Engineering, Southwest Jiaotong University. His research interests include industrial robot, machine vision and visual servoing.

Xinliang Yao received the B.S. degree in Mechanical Engineering from Northeast Forestry University, Harbin, China, in 2018. He is currently a postgraduate in the School of Mechanical Engineering, Southwest Jiaotong University. His research interests include deep learning, object detection and visual servoing.