Visual Servo Control of Industrial Robot Based on Convolutional Neural Network

Quanfang Li¹, Liucun Zhu²*, Boquan Li¹, Xiaodong Zheng² and Junqi Luo²

¹School of Mechanical Engineering, Jiangsu University, Zhenjiang, China
²School of Advanced Science and Technology Research Institute, Beibu Gulf University, Qinzhou, China

*Corresponding author e-mail: lczhu@bbgu.edu.cn

Abstract. Aiming at the problem of poor flexibility of traditional industrial robot, the vision system and industrial robot control system are combined to propose a visual servo control method based on Faster RCNN. The image information of the target object is collected through the camera, and the Convolutional Neural Network (CNN) is used to detect the entire picture, then the category and position of the object to be grasped are obtained, which improves the control performance of the robot system. Experiments show that the visual servo robot control system can quickly complete the position recognition of different objects. Mean average precision of this method can reach 0.867. The maximum error between the recognized grasping position and the actual position on the coordinate axis does not exceed 3.4 mm, the detection time of each picture in the GPU environment is 17.1 ms, which has certain guiding significance in the visual servo control of industrial robots.

1. Introduction

At present, in the visual servo grasping tasks of industrial robots, traditional feature extraction methods are mainly used [1], such as the extraction of features such as color, texture, and spatial relationship [2]. This method is manually calibrated by the operator, and is affected by changes in the posture of the target and the lighting of the environment, and it is difficult to adapt to unknown objects or when working in an unstructured environment [3-5]. The neural network method is applied to the robot's grasping task, so that the robot has a more efficient perception of objects, and then the robot has a certain intelligent ability [6-8]. In recent years, deep learning technology has developed rapidly. Neural networks represented by convolutional networks have made major breakthroughs in machine vision, language recognition, unmanned driving, health monitoring and many other fields [9-10]. Compared with the process of artificially designing complex features, the use of convolutional neural networks can train large amounts of data to extract feature expressions suitable for the current project. Convolutional neural networks involve many parameters, mainly because convolutional networks must stack many layers in order to improve feature expression capabilities [11]. Therefore, compared with traditional robot control algorithms, more annotation information is required for feature expression training, which improves the generalization of the algorithm ability. With the advent of the big data era, it has become very easy to use information processing technology to obtain a large amount of training data, and the research of convolutional neural networks has continuously made breakthroughs [12-15]. The convolutional neural network is used to process the image information.
collected by the camera, identify the type and grasp the object, and control the robot to use different claw poses and grasping forces to grasp.

2. Robot visual servo
The visual servo robot processes the image information through the vision system, feeds it back to the robot system, and controls the robot's movement according to the feedback information. The main purpose of image information processing is to calculate the position of the object in the image in the world coordinate system, which involves the conversion of the coordinate system, that is, how to transform the pixel coordinates of the image into the world coordinate system.

2.1. Visual Servo
In the academic field and industrial production field, the visual servo system is often divided into position based visual servoing system, image based visual servoing system and homography based visual servoing system according to the different feedback signals set in the objective function. The image-based visual servo system directly uses the camera to obtain image information as the feedback information of the entire system, and compares the existing image feature $f$ with the given image feature $f^*$ to generate a control signal to achieve the closed-loop control of the entire system. Figure 1 shows the image-based visual servo system.

![Figure 1. The image based visual servoing system](image)

The system structure is simple, and the control algorithm calculation is small, so it is widely used in industrial production.

2.2. Vision-based target positioning principle
Before the robot grasps, it needs to know the six-degree-of-freedom pose of the target in the robot coordinate system. When using the camera to obtain the position information of the target object, the points in the image need to be converted to the robot coordinate system. The whole conversion process involves the conversion between the image coordinate system, the camera coordinate system and the world coordinate system. As shown in Figure 2 coordinate system conversion.

![Figure 2. Coordinate system conversion](image)

The mathematical relationship between the world coordinate system and the camera coordinate system is:

$$
Z = \begin{bmatrix}
\frac{1}{u} \\
\frac{1}{v} \\
1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & u_0 \\
0 & 1 & v_0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
f & 0 & 0 & 0 \\
0 & f & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
R \\
T \\
1
\end{bmatrix}
\begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix}
$$

(1)
In the above formula (1), $f$ is the focal length of the camera, $d_x$ and $d_y$ respectively represent the physical size of the horizontal and vertical coordinates of each pixel in the image coordinate system, $u_0$ and $v_0$ represent the origin of the image coordinate system, $R$ and $T$ represent the rotation matrix and the translation matrix respectively, $Z_c$ is the image coordinates, $(u,v)$ the corresponding depth value.

3. Faster RCNN
The Faster RCNN algorithm is proposed by He Kaiming et al\cite{16}. It is one of the most widely used image detection algorithms. Compared with the two versions of Fast RCNN and RCNN, Faster RCNN has faster detection speed and higher detection accuracy.

3.1. Faster-RCNN processing process
Faster RCNN continues the idea of RCNN. It first generates the Region Of Interest (ROI), then classifies the generated region, and finally completes the object detection\cite{17-18}. The ROI is generated using the Region Proposal Network (RPN), and the region classification is the RCNN\cite{19}.

In terms of functional module, it mainly includes four parts: feature extraction, RPN, ROI Pooling and RCNN. The input image first through the feature extraction network to obtain a feature map, and generates a better suggestion frame through the RPN, and uses the Anchor\cite{20}.

The RPN includes Anchor box, RPN convolutional network, RPN loss, Proposal and ROI; the RPN convolutional network corresponds to the anchor box, and each point in the generated feature map corresponds to 9 anchor boxes. Therefore, the convolution can be used to generate the prediction score and the prediction offset value of the corresponding anchor frame on the feature map. At this time, there are still a lot of suggested boxes, and the suggestion boxes need to be further filtered to get the final ROI. Using the predicted score and offset of each anchor box, a better set of suggested boxes is further obtained. But in the testing phase, the suggested box is directly used as the area of interest. Since the RCNN module uses a fully connected network and requires a fixed feature dimension, ROI Pooling is used to pool the features of the ROI to a specific dimension. The RCNN is responsible for sending the features obtained by ROI Pooling to the fully connected layer, predicting the classification of each ROI, predicting the offset to correct the position of the bounding box, calculating the loss, and completing the entire Faster RCNN process.

3.2. Grab location recognition
For object detection tasks, it is necessary to predict the category of each object and its grasping position, that is, category, center coordinate $x$ and $y$, width $w$ and height $h$.

The regional candidate network RPN uses the category of the predicted anchor frame as the category of the predicted frame, and can predict the offset of the real frame relative to the anchor frame, instead of directly predicting the center point coordinates $x$ and $y$, width $w$ and height $h$.

![Figure 3. The relationship between the anchor box and the label](image)

As shown in the relationship between the anchor box and the label in Figure 3, there are three anchor boxes and two labels in the input image. From the position point of view, the anchor boxes A and C overlap with the labels M and N to a certain extent, and The position of the anchor frame B is the...
background. The RPN determines whether an anchor frame belongs to the foreground or the background by calculating the IOU of the anchor frame and the label. The IOU is the ratio of the common part of the two boxes to all parts, that is, the overlap ratio. The calculation formula for the IOU of the anchor frame A and the label M is shown in equation (2).

$$IOU(A, M) = \frac{A \cap M}{A \cup M}$$  \hspace{1cm} (2)

When the IOU is greater than a certain value, the true value of the anchor frame is the foreground, and when it is lower than a certain value, the true value of the anchor frame is the background. When solving the true value of the offset, for anchor frame A and label M, suppose that the center coordinates of anchor frame A are $x_a$ and $y_a$, the width $w_a$ and height $h_a$ respectively, the center coordinates of label M are $x$ and $y$, width and height are $w$ and $h$ respectively. Then the corresponding true value of the offset can be calculated by formula (3).

$$
\begin{align*}
t_x &= \frac{(x - x_a)}{w_a} \\
t_y &= \frac{(y - y_a)}{h_a} \\
t_w &= \log(\frac{w}{w_a}) \\
t_h &= \log(\frac{h}{h_a})
\end{align*}
$$  \hspace{1cm} (3)

It can be seen from the above formula that the position offset is normalized with width $t_x$ and height $t_y$, the width $t_w$ and height $t_h$ offset is logarithmically processed to further limit the range of the offset and facilitate prediction. After that, the RPN obtains the predicted value of the category and the offset through the convolutional network, which is $t_x^*, t_y^*, t_w^*$ and $t_h^*$.

After the prediction offset is obtained, the prediction offset is applied to the corresponding anchor frame to obtain the actual position $x^*$, $y^*$, $w^*$ and $h^*$. The calculation process is shown in equation (4).

$$
\begin{align*}
t_x^* &= \frac{(x^* - x_a)}{w_a} \\
t_y^* &= \frac{(y^* - y_a)}{h_a} \\
t_w^* &= \log(\frac{w^*}{w_a}) \\
t_h^* &= \log(\frac{h^*}{h_a})
\end{align*}
$$  \hspace{1cm} (4)

4. Experiment and analysis

In order to verify the feasibility of the algorithm, five kinds of objects were collected by the camera: comb, cup, plier, puppet mushroom and puppet hedgehog. Record a 2-minute video in 3 different environments, and 1 second includes 25 frames of video. After the 2-minute video is processed to generate 200 pictures, for a total of 3000 pictures. Among them, 2100 pictures are used as the train set and 900 pictures are used as the test set.

In order to verify the performance of the algorithm, three algorithms of Faster RCNN, YOLOv1 and Improved Autoencoder are used as comparative experiments, and the image data collected by themselves are used for training. The experimental results are shown in Figure 4.
From the above analysis, the mean average precision based on Faster RCNN can reach 0.867, and the mean average precision based on YOLOv1 algorithm can reach 0.816. The Improved Autoencoder, the highest average value is only 0.676, which proves that the detection accuracy of Faster RCNN algorithm is higher.

The actual coordinates of the center point of the object grabbing position can be tested by the measuring instrument, and the coordinates of the object grabbing center point identified by the Faster RCNN algorithm can be calculated to obtain the specific values of the axis and the axis error, as shown in Figure 5. Error with the actual point position on the x, y axis is shown.

![Figure 5](image)

Figure 5. Error with the actual point position on the x, y axis

It can be seen from Figure 3-3 that the position error range of the grasping point in the axial direction is 1.2-3.2mm, and the position error range in the axial direction is 1.6-3.4mm.

5. Conclusion

This paper proposes to use Faster RCNN to detect grasped objects and the grasped positions. Experiments show that the average detection accuracy of this algorithm is relatively high. Compared with the YOLOv1 algorithm, the average average accuracy is 4.4% higher, and the recognized grasping position does not exceed the actual position error by 3.4mm. Experiments show that the deep learning method based on Faster RCNN can better extract the location information of the detected target. Compared with the previous detection method, the performance of the detection method can be improved. Since convolutional networks require a lot of calculations, the real-time processing capabilities of the entire system must be considered in practical applications, so experiments will be carried out on the robot system next.

6. Acknowledgments

This work was financially supported by Guangxi Bagui Scholars Program fund.
References

[1] Dan Kielsholm Thomsen, Rune Søe-Knudsen, Ole Balling, et al. Vibration control of industrial robot arms by multi-mode time-varying input shaping. 2021, 155

[2] Maike Ketelhut, Melanie Kolditz, Fabian Göll, et al. Admittance Control of an Industrial Robot during Resistance Training. 2019, 52(19):223-228.

[3] Shuaipu Jin, Xiufeng Wang, Leilei Du, et al. Evaluation and modeling of automotive transmission whine noise quality based on MFCC and CNN. 2021, 172

[4] Cen Han, Guangyu Gao, Yu Zhang. Real-time small traffic sign detection with revised faster-RCNN. 2019, 78(10):13263-13278.

[5] Fan Xu, Hesheng Wang, Weidong Chen, et al. Adaptive visual servoing control for an underwater soft robot. 2018, 38(5):669-677.

[6] Xudong Sun, Pengcheng Wu, Steven C.H. Hoi. Face detection using deep learning: An improved faster RCNN approach. 2018, 299:42-50.

[7] Pau Muñoz-Benavent, Luis Gracia, J. Ernesto Solanes, et al. Sliding mode control for robust and smooth reference tracking in robot visual servoing. 2018, 28(5):1728-1756.

[8] Guoyang Wan, Fudong Li, Guofeng Wang. Visual Positioning and Grasping Application of Industrial Robot for Casting Parts. 2019, 7(1)

[9] Yun Hong Liu, Hao Yang Yu. Visual Servo Robotic Arm Control System Based on LabVIEW. 2014, 3365:844-847.

[10] Yue Guo, Xiaolan Yao, Wei Li. Design of Visual Servo Robot Tracking System Based on Image Feature. 2015,

[11] Nan Mo, Li Yan. Improved Faster RCNN Based on Feature Amplification and Oversampling Data Augmentation for Oriented Vehicle Detection in Aerial Images †. 2020, 12(16)

[12] Zhang Jinpeng, Zhang Jinming, Yu Shan. Hot Anchors: A Heuristic Anchors Sampling Method in RCNN-Based Object Detection.. 2018, 18(10)

[13] Chaojie Li, Peng Zhou. Improved Faster RCNN Object Detection. 2020, 6(3)

[14] Xinwen Gao, Ming Jian, Min Hu, et al. Faster multi-defect detection system in shield tunnel using combination of FCN and faster RCNN. 2019, 22(13):2907-2921.

[15] XiPeng Yu, Zhong Chen, Shuo Zhang, et al. A street rubbish detection algorithm based on Sift and RCNN. 2018, 10608:106080F-106080F-8.

[16] Tim Haas, Christian Schubert, Moritz Eickhoff, et al. BubCNN: Bubble detection using Faster RCNN and shape regression network. 2020, 216

[17] Wafaa M. Salih Abedi. Unconsciousness detection supervision system using faster RCNN architecture. 2018, 1-6.

[18] Bian Yingxian, Guo Genchang. Comparative Analysis of Performance Using Faster RCNN and ACF in People Detection. 2017, 15(6):11-21.

[19] Hafiz Tayyab Mustafa, Masoumeh Zareapoor, Jie Yang. MLDNet: Multi-level dense network for multi-focus image fusion. 2020, 85

[20] Cong Feng, Jie Zhang. SolarNet: A sky image-based deep convolutional neural network for intra-hour solar forecasting. 2020, 204:71-78.