Mobile Robot Object Recognition in The Internet of Things based on Fog Computing

Meixia Fu*, Songlin Sun*, Kaili Ni*, Xiaoying Hou*
*National Engineering Laboratory for Mobile Network Security, Beijing University of Posts and Telecommunications
*Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing University of Posts and Telecommunications
*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications
{mxfu, slsun}@bupt.edu.cn

***Abstract*—Mobile robot object recognition has attracted significant attention in the internet of things recently, in which there are many challenging tasks, such as the objects, the communication networks and the computer system. It still needs a large of computation, communication and storage capability for the whole system. In this paper, we propose a scheme of mobile robot object recognition in IOT and use edge nodes to process the data from robot vision instead of cloud computing. Besides, we adopt YOLOv3 as the main algorithm in the edge nodes to process the video data. The video from the camera on the robot is transmitted to the fog node by a Wi-Fi router. The advantages of using edge nodes for computation local robot clusters are reliability, real-time capability and flexibility compared with the cloud. In our experiment, we efficiently train the computer model using COCO database deployed to GPU. The robot using the proposed model could recognize the objects in real-time. We achieve mAP of 31.0% and response time of 52ms that illustrates the state-of-art performance of the proposed framework.**

***Index Terms*— Robot object recognition, The internet of things, Fog computing**

I. INTRODUCTION

The Internet of Things (IoT) has attracted tremendous attention from the industry and academia with the rapid development of modern wireless communication system, specifically the fifth generation (5G) system. It aims that universal things - such as electronics, vehicles, buildings, physical objects and other items - could access to the Internet anywhere and anytime though Radio-Frequency IDentification (RFID), sensors, actuators, mobile devices, etc [1-3]. Intelligent IoT enables a variety of applications and services both human-objects and objects-objects, which bring dramatic benefits to our life [4]. IoT has three main components that are the objects, the communication networks and the computer system that process the data streaming of the object transmitted by the communication networks [5]. The communication networks and the computer system are challenging for intelligent robot real-time recognition in IoT.

Many methods of mobile robot object recognition have been reported in recent years. Browatzki *et al.* [6] adopted Pyramids of Histograms of Oriented Gradients feature (PHOG) to capture the shape information and color histograms to describe the object appearance that used in robot object recognition. Cartucho *et al.* [7] proposed a new neural network called HHELP based on YOLOv2 [8] and use the data collected from the realistic scene for the mobile intelligent robot. Moreover, there are many popular algorithms of object recognition in computer vision [9-11] which obtain different performance on speed and accuracy. Here, we use YOLOv3 as the main algorithm in the computer system to attain real-time recognition [11].

In order to recognize the real-time video images, the whole system needs a large of communication, computation, and storage capability. In some prior work, many researchers use cloud computing technology for better computation and storage capability [12]. However, the whole process needs to be completed on the cloud, which could occupy extensive network bandwidth. Meanwhile, it will cause network delay, service interruption, network security and other issues, even if many approaches have been reported to optimize the network transmission [5].

Fog computing is a novel computational model. It is considered as the extension of cloud computing from the core of the network to the edge of the network [13]. The purpose of fog computing is to storage, process and delivery data from the terminal devices to the cloud servers. In fog computing, fog nodes mean the facilities or infrastructures that can offer resources for services at the edge of the network. Most fog nodes can also be responsible for controlling and monitoring the other components inside the node, such as storage, accelerators, et al. The computing power and data storage in fog computing are closer to the end devices. What's more, the network transmission and delay are dramatically decreased compared with cloud computing [14].

In this paper, a object recognition system based on YOLOv3 is presented to locate and recognize the objects in robot vision, which could process per image over Human-level accuracy. A fog computing with object recognition framework is proposed to let the robot percept and cognize the around things like human beings. This can reduce the network latency and the requirement of bandwidth. Besides, it is practical and meaningful that will be applied into many applications, such as smart home, self-driving, intelligent surveillance and human-machine interaction. The remainder of this paper is introduced

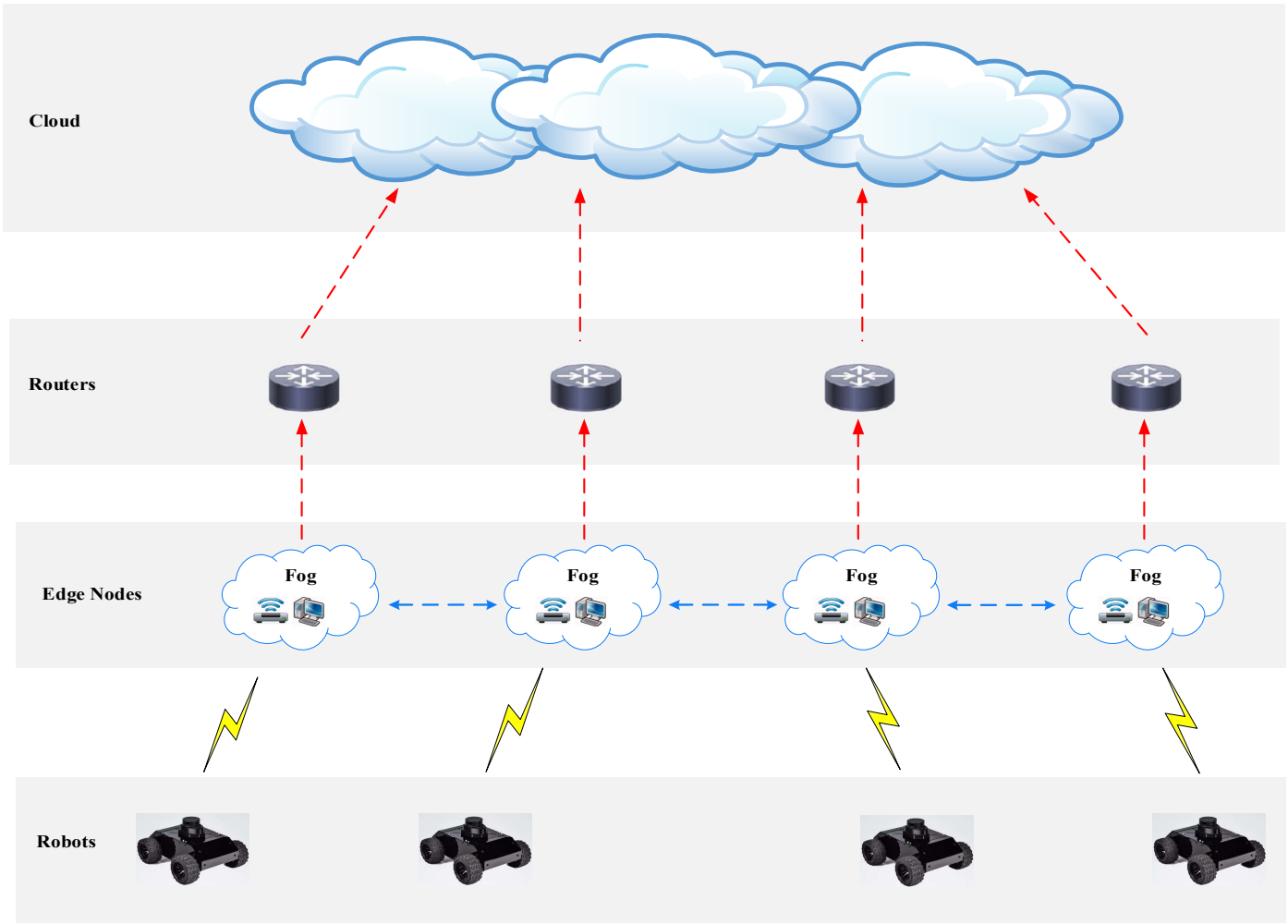


Figure 1 The scheme ofintelligent robots accessed in IoT based on fog computing.

as follows. Section II presents the proposed model in detail. Section III verifies the state-of-the-art performance of the proposed framework. Section IV gives a conclusion of this paper and put forward some research direction about the future work.

II. THE PROPOSED MODEL

The architecture of fog computing based on mobile robot object recognition is shown in Figure 1. This scheme consists of three main parts: terminal devices, fog and cloud. Terminal devices contain electronic equipment like computers, mobile phones, robots and other devices. In the practical, we only use one robot in the terminal. Fog always consists of many fog nodes. Cloud contains management server, resolution server, information server, and data center. Each module is introduced concretely as follows:

1) Terminal devices: The terminal robot with cameras is responsible for collecting visual information that will be transmitted to the fog node by a Wi-Fi-router. Here, we apply an open-source Robot Operating System (ROS) [15] into the robot that provides different capabilities like navigation, manipulation, control. So, the robot could walk and bypass obstacles according to the orders.
2) Fog: Fog consists of many fog nodes that contain the common facilities (such as a Wi-Fi router, a computer, GPUs), which can provide services for the robot at the edge of the network. The fog nodes not only are responsible for delivering services to the cloud by routers but also offers computation capability to process the video data from the terminal. What's more, it returns some orders to the robot, for example, when the robot finds one book on the ground, it will put the book on the table.
3) Cloud: The management server connects with fog nodes for various IoT requirements. Resolution server and information server are responsible for scheduling resources and allocating computing tasks. Meanwhile, it returns the processed information to the fog nodes.

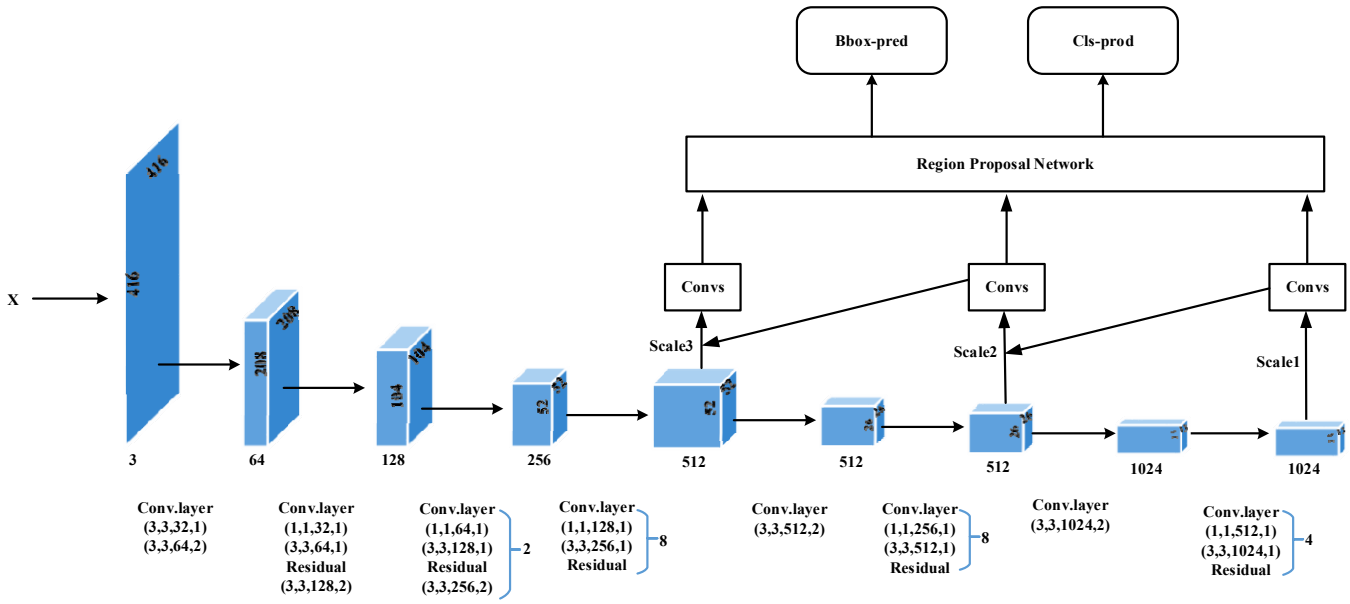


Figure 2 The system of object recognition based on YOLOv3.

III. THE MAIN ALGORITHM OF PROCESSING DATA

A. The system of object recognition based on YOLOv3

In order to complete the real-time recognition, we adopt YOLOv3 as the main algorithm to process the radio from the robot, as shown in Figure 2. This model is composed of several convolutional layers and residual blocks for extracting features. The original image is resized to (416, 416) as the input of this structure. Darknet-53 is used in this model as the feature extraction network, which achieves obvious performance in object detection [11]. It has 53 convolutional layers, some of which compose residual blocks [16]. In a convolutional layer, the size of convolutional kernel is like (3, 3) and (1, 1) that are the first and second parameter in (3, 3, 32, 1) or (1, 1, 32, 1). The third in (3, 3, 32, 1) is the filter number in one layer and the last one is convolution stride. The remarkable thing is to double the channels and half the size by one convolutional layer with stride 2 rather than one convolutional layer with stride 1 and one pooling layer with the kernel (2, 2). Besides, Darknet-53 adds batch normalization [17] before the convolutional layers for increasing the model convergence speed. What's more, it adopts three-scale feature maps of (13, 13), (26, 26) and (52, 52) to alleviate that the feature of tiny things in the picture are easily vanished after deep convolutional layers [18].

After feature extraction, we get the feature map at size of S ×S. If the center of an object in the image falls into one of the S × S grid cell, that cell is responsible for detection and classification. Each grid cell in the last feature map predicts B bounding boxes. These predictions are encoded as an S×S×B×(4+1+C). Here, C are the amount of classes in the bounding box, separately. Darknet-53 uses three scales to predict boxes. Each cell in the last feature map predicts B/3 bounding boxes at each scale.

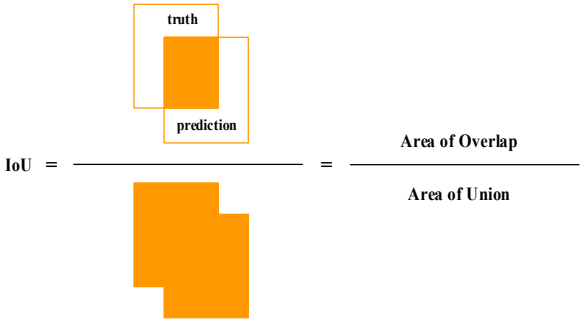


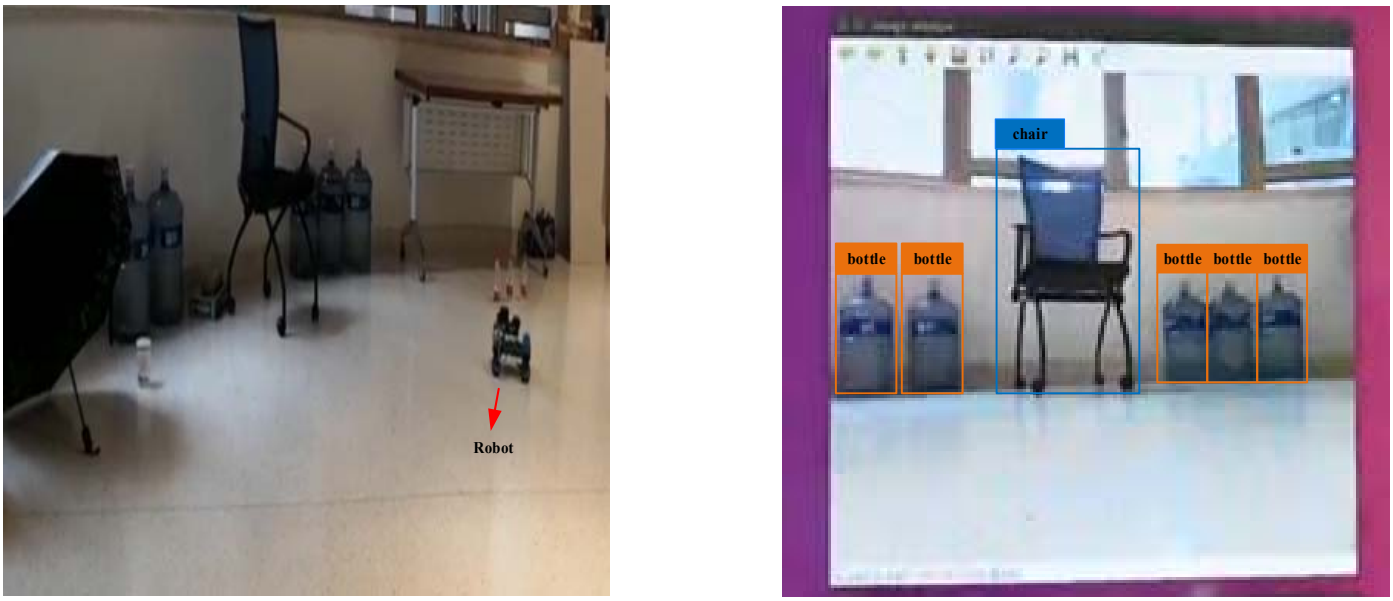
Figure 3 Intersection-over-Union.

B. Bounding box prediction

The output dimension of each bounding box is 5 coordinates $(t_x, t_y, t_w, t_h, t_o)$ for detection and classification. Here, t_x is the offset between the bounding box and truth box. (t_y, t_w, t_h) is as well as t_x. The 5 coordinates stabilize to predict the offsets after a transmission of a sigmoid function. $(c_x, c_y)$is the cell offset from the top left corner in an image. Besides, (p_w, p_h) is the width and height value of the bounding box. After regression, the function of the predictions are as follows:

$$b_x = \delta(t_x) + c_x \quad (1)$$
$$b_y = \delta(t_y) + c_y \quad (2)$$
$$b_w = p_w e^{t_w} \quad (3)$$
$$b_h = p_h e^{t_h} \quad (4)$$
$$b_o = logistic\ Regression(t_o) \quad (5)$$

(b_x, b_y, b_w, b_y) are donated to loss function. t_o is the confidence score that presents the probability of object in the



(a) **(b)**

Figure 4 The experimental result of object recognition on the robot in IoT based on fog computing. Figure 4(a) shows the universal things around the robot. Figure 4(b) presents the position and classification of the things that the robot sees.

bounding box. That is sent to logistic regression function to predict an object score of each bounding box. If there is no object in this cell, t_o is 0.

C. MULTIPLE-TASK TRAINING

The whole system has two tasks, object detection and classification. When obtaining the feature maps, we predict the position for each cell in the feature map. The overall output is S×S×B×(D+C). The average Intersection-over-Union (IoU) is the evaluation criterion. Figure 3 shows that IOU is ration between the intersection of truth box and the predictive box in the same image and the union of the truth box and the predicted box. The value of IoU is limited to (0, 1). With the bigger of reduce redundancy, non-maximum suppression is applied to limit the proposals with the IoU threshold at 0.5.

The multi-mask loss function includes detection and classification. We use sum-squared error to optimize the models. The multi-mask loss function is

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}[(b_{xi}-\hat{b}_{xi})^2+(b_{yi}-\hat{b}_{yi})^2]$$

$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}[(b_{wi}-\hat{b}_{wi})^2+(b_{hi}-\hat{b}_{yi})^2]$$

$$+\lambda_{obj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[(b_{oi}-\hat{b}_{oi})^2\right]$$

$$+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{noobj}[(b_{oi}-\hat{b}_{oi})^2]$$

$$+\lambda_C\sum_{i=0}^{S^2} I_i^{obj}\sum_{c\in classes}(p_i(c)-\hat{p}_i(c))^2 \qquad (6)$$

Here, I_i^{obj} means that the proposal appears in cell i, I_{ij}^{obj} means that the *jth* bounding box in cell i is responsible for loss function optimization. The loss function only penalizes bounding box that has the highest IOU of any predictor in that grid cell. We set $\lambda_{coord}=1$, $\lambda_{obj}=5$, $\lambda_{noobj}=1$ to increase the effect of bounding box that are responsible to detection. $p_i(c)$ means the prediction accuracy of the object in each cell and $\hat{p}_i(c)$ is the truth prediction accuracy. In order to decrease the effect on the overall loss function, we set $\lambda_C=0.5$.

IV. EXPERIMENTAL RESULTS

In this section, we collect the radio data from the robot and transmit it to the fog node by a Wi-Fi router. Here, we don't establish a connection between the fog node ad the cloud. So, there is only one connection between the robot and the fog node. The computer system in the fog node is responsible for processing data, the main algorithm of which is based on YOLOv3. The proposed framework is deployed on NVIDIA GeForce GTX 1080Ti.

In training, the proposed scheme based on YOLOv2 and YOLOv3 are trained on COCO dataset [19], the classes of which can reach 80 for the objects. We randomly resize the original images sent to the model every 10 batches. The new image dimension size includes {320, 352, …, 416} with the model down-samples by a factor of 32. The network can choose one new dimension size and continue to be trained after every 10 batches.

Table 1 Comparison of our scheme with YOLOv2 and YOLOv3 for COCO database. The rank accuracy and response time are listed.

Model	mAP	Response time
Our scheme _ YOLOv2	21.6%	87ms
Our scheme_ YOLOv3	31.0%	52ms

In Table 1, we can see the proposed scheme based on YOLOv3 significantly achieves the state-of-art performance with mAP of 31.0% and response time of 52ms, which are respectively 9.4% and 35ms improvements compared with the YOLOv2. Figure 4 shows us the test interface of mobile robot object recognition in IoT based on fog computing. Figure 4(a) presents the practical environment with universal things around the robot. Besides, the robot runs according to the orders. Figure 4(b) shows the test result about the position and class of the things that the robot sees. Consequently, the experimental results verify the state-of-art performance of the proposed scheme. We could complete the real-time recognition for the robot in IoT, that is meaningful and practical for other tasks, like action recognition, behavior prediction, and human-machine interaction.

V. CONCLUSIONS

This paper presents an effective scheme to address mobile robot object recognition based on fog computing in IoT. We adopt YOLOv3 as the main algorithm in the computer system to process the video data from the robot. Besides, we also deploy the processing to GPU and complete the real-time performance. In the future, we intend to research the influence of the fog nodes, the cloud and the mutual connection on the object recognition. What's more, we will continue to do some research about other tasks in computer vision based on object recognition.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (Project61471066) and the open project fund (No.201600017） of the National Key Laboratory of Electromagnetic Environment, China.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Comput. Netw., vol. 54, no. 15, pp. 2787–2805, 2010.
[2] Al-Fuqaha A, Guizani M, Mohammadi M, et al. Internet of things: A survey on enabling technologies, protocols, and applications[J]. IEEE communications surveys & tutorials, 2015, 17(4): 2347-2376.
[3] Hu P, Ning H, Qiu T, et al. Fog computing based face identification and resolution scheme in internet of things[J]. IEEE transactions on industrial informatics, 2016, 13(4): 1910-1920.
[4] Intelligent Device Discovery in the Internet of Things - Enabling the Robot Society.
[5] Stergiou C, Psannis K E, Kim B G, et al. Secure integration of IoT and cloud computing[J]. Future Generation Computer Systems, 2018, 78: 964-975.
[6] Browatzki B, Tikhanoff V, Metta G, et al. Active object recognition on a humanoid robot[C]. 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012: 2021-2028.
[7] Cartucho J, Ventura R, Veloso M. Robust Object Recognition Through Symbiotic Deep Learning In Mobile Robots[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 2336-2341.
[8] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[J]. 2016.
[9] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]. International Conference on Neural Information Processing Systems. MIT Press, 2015:91-99.
[10] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[C] European Conference on Computer Vision. Springer, Cham, 2016:21-37.
[11] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. 2018.
[12] Beksi W J, Spruth J, Papanikolopoulos N. Core: A cloud-based object recognition engine for robotics[C]. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015: 4512-4517.
[13] Bonomi F, Milito R, Zhu J, et al. Fog computing and its role in the internet of things[C]. Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012: 13-16.
[14] M. Hajibaba and S. Gorgin, "A review on modern distributed computing paradigms: Cloud computing, jungle computing, and fog computing," J. Comput. Inf. Technol., vol. 22, no. 2, pp. 69–84, 2014.
[15] Quigley M, Conley K, Gerkey B, et al. ROS: an open-source Robot Operating System[C]//ICRA workshop on open source software. 2009, 3(3.2): 5.
[16] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015:770-778.
[17] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift[J]. 2015:448-456.
[18] Lin T Y, Dollár P, Girshick R, et al. Feature Pyramid Networks for Object Detection[J]. 2016:936-944.
[19] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollr, P. and Zitnick, C.L., 2014. "Microsoft COCO: Common Objects in Context." In ECCV, (pp. 740-755). Springer.