2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI) | 978-1-6654-3208-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CCCI52664.2021.9583186

Real-Time Instance Segmentation for Low-Cost Mobile Robot Systems Based on Computation Offloading

Yuanyan Xie
University of Science and Technology Beijing
Beijing 100083, China
yyxie@xs.ustb.edu.cn

Yu Guo*
University of Science and Technology Beijing
Beijing 100083, China
Shunde Graduate School of University of Science and Technology Beijing
Foshan 528399, China
guoyu@ustb.edu.cn

Yue Chen
University of Science and Technology Beijing
Beijing 100083, China
chenyue@xs.ustb.edu.cn

Zhenqiang Mi
University of Science and Technology Beijing
Beijing 100083, China
mizq@ustb.edu.cn

***Abstract*—Instance segmentation can enable mobile robots to obtain the environmental semantic information and accomplish more complex interaction with environments, such as navigation, grasping, and virtual reality. However, low-cost mobile robots have limited onboard resources, and can not afford the massive computation of instance segmentation methods. This paper proposes a real-time instance segmentation framework for mobile robot systems based on computation offloading, which offloads part of computation of the instance segmentation network to the cloud, and leverages the powerful computation resources and sufficient memories on the cloud platform to accelerate the network. First, we formulate the instance segmentation network as the directed acyclic graph, and present its time cost model and energy consumption model. Then, a computation offloading strategy is proposed to reduce the time cost of the whole instance segmentation and the energy consumption on the mobile robot. Our framework has been verified on the representative one-stage method, Yolact, and two-stage method, Mask R-CNN. The results show that our framework can accelerate the execution of instance segmentation network on mobile robots, and achieve the speed of around one second per frame.**

***Index Terms*—instance segmentation, mobile robot, semantic understanding, computation offloading**

I. Introduction

Instance segmentation, which is one of the most challenging computer vision tasks, aims to detect the object instances in an image while localizing them with both bounding boxes and segmentation masks, as shown in Fig. 1. Due to its capability of understanding the environments through the vision, instance segmentation has a wide range of application scenarios and also attracts increasing researchers from the field of robotics. As the complement of geometrical robot perception methods, instance segmentation can enable robots to obtain the high-level semantic informantion of environments, and to accomplish more complex interaction with environments, such as navigation, grasping and so on.

Fig. 1. Instance segmentation results on the COCO test set. Masks are shown in color, and bounding boxes, classes, and confidences are also shown.

Classical instance segmentation methods have two categories: two-stage methods [1] and one-stage methods [2], [3]. Two-stage methods first generate candidate region-of-interests (ROIs), and then classify and segment those ROIs in the second stage. This kind of two-stage structures severely limits their speed, and makes them far from real-time. One-stage methods discard the time-consuming region proposal step, and break instance segmentation into two parallel subtasks: generating prototype masks and predicting mask coefficients. The state-of-the-art one-stage methods achieve real-time performance, but they still need to be accelerated with high-performance graphics processing unit (GPU).

However, mobile robots have limited onboard resources including computation, memory, and battery capacity in some

978-1-6654-3208-5/21/$31.00 ©2021 IEEE

specific scenarios, such as disaster rescue, forest protection, and house keeper, because they need to move in a narrow space and their volume and weight should be minimized. Moreover, high-performance processors are quite expensive and impede the wide spread of mobile robots in daily life. Therefore, these kind of mobile robots can not afford the massive computation and intensive data of the instance segmentation methods.

To address this issue, some researches compress the network with parameter pruning and sharing, low-rank factorization, transferred/compact convolutional filters, knowledge distillation, or other related techniques [4]. However, these methods will lead to the significant decrease of accuracy. Other attempts directly upload the raw data collected by sensors to the cloud server and all computation will be finished in the cloud. Nevertheless, the data transmission from robot to cloud is also time-consuming and has a risk of data leakage.

In this paper, we propose a real-time instance segmentation framework for low-cost mobile robot system, which offloads part of computation of the instance segmentation network into the cloud. It is inspired by our previous works [5]–[7], which also utilize the cloud resources to extend the capability of robots. In this way, mobile robots can utilize the powerful cloud resources to accelerate the execution of network. At the same time, mobile robots will also process the data to extract the useful features and discard the redundancy information, so that the data transmission time can be reduced and the data security can be enhanced.

The rest of this paper is organized as follows. Section II presents our system model and gives the problem formulation. Based on the proposed model, a real-time instance segmentation framework based on computation offloading is proposed in Section III. The experimental setups and results are reported in Section IV. Finally, Section V concludes the paper and presents our future work.

II. System Model and Problem Formulation

In this paper, we assume a scenario that a low-cost mobile robot is not equipped with high-performance computing processors and has limited battery capacity. But fortunately, it can communicate with the remote cloud server, and utilize the numerous resources on the cloud platform, including high-performance computing processors, graphics processing units, and a mass of memory. Our goal is to deploy the instance segmentation network on this kind of mobile robots and achieve real-time preformance, enabling these robots to understand the environments with semantic information (e.g. what object instances there have and where they are).

The instance segmentation network can be modelled as the directed acyclic graph $\mathcal{G} =< \mathcal{M}, \mathcal{D} >$. The nodes $\mathcal{M} = \{m_1, m_2, ..., m_K\}$ in the graph represent a series of modules in the network, including the backbone network which is used to extract the basic features and various functional modules to realize particular classification or segmentation tasks. Furthermore, the backbone network can be broken into multiple convolutional layers, and each layer can be regarded as an independent module. Note that all modules must be sequential. Two components which are parallel in the instance segmentation network, should be merged as one module. The edges $\mathcal{D} = \{d_{ij} = (m_i, m_j) | m_i, m_j \in \mathcal{M}\}$ in the graph denote data transmission among different modules, and the length of edge $|d_{ij}|$ represents the amount of data delivered from m_i to m_j.

A. Time Cost Model

In the aforementioned directed acyclic graph, each module can be deployed on the mobile robot or cloud platform, and its execution time depends on the hardware configuration (e.g. CPU, GPU, and memory). Usually, for the same module m, the execution time on a mobile robot $t_r(m)$ is much longer than that on a cloud platform $t_c(m)$. There are two types of data transmission mode: Robot-to-Robot transmission (R2R) and Robot-to-Cloud transmission (R2C). We assume that the data transmission time in the R2R mode can be ignored, i.e., $t(d, R2R) = 0$, and the data transmission time in the R2C mode depends on the amount of data and the communication bandwidth, denoted as $t(d, R2C, B)$, where B represents the communication bandwidth.

The time cost of the whole instance segmentation network includes both module execution time t_{exec} and data transmission time t_{trans}, and can be represented as (1).

$$t_{total} = t_{exec} + t_{trans} \tag{1}$$

We consider three situations: 1) execute part of modules on the cloud; 2) execute all modules on the cloud; 3) execute all modules on the mobile robot.

In the first case, the module execution time can be calculated by (2), where $\mathcal{M}_R$ represents the set of modules executed on the mobile robot, and $\mathcal{M}_C$ denotes the set of modules executed on the cloud platform. The data transmission time can be calculated by (3), where d is the intermediate results through the processing of modules on the mobile robot. Consequently, the total time cost can be represented as (4).

$$t_{exec} = \sum_{m \in \mathcal{M}_R} t_r(m) + \sum_{m \in \mathcal{M}_C} t_c(m) \tag{2}$$

$$t_{trans} = t(d, R2C, B) \tag{3}$$

$$t_{total} = \sum_{m \in \mathcal{M}_R} t_r(m) + \sum_{m \in \mathcal{M}_C} t_c(m) + t(d, R2C, B) \tag{4}$$

In the second case, the module execution time can be calculated by (5), where $\mathcal{M}$ includes all modules in the instance segmentation network. The data transmission time can also be calculated by (3), but d refers to the raw image or video data obtained from sensors. Therefore, the total time cost can be represented as (6).

$$t_{exec} = \sum_{m \in \mathcal{M}} t_c(m) \tag{5}$$

$$t_{total} = \sum_{m \in \mathcal{M}} t_c(m) + t(d, R2C, B) \tag{6}$$

In the last case, the whole instance segmentation network is executed on the mobile robot, and the total time cost can be represented as (7).

$$t_{total} = \sum_{m \in \mathcal{M}} t_r(m) \tag{7}$$

B. Energy Consumption Model

This paper merely considers the energy consumption on the mobile robot. This is because the battery capability of mobile robots is limited, and the energy consumption will significantly influence the robots' standby time. On the contrary, cloud platforms have continuous power supply, and can operate all the time no matter how much energy tasks consume.

During the execution of instance segmentation network, the energy consumption on the mobile robot consists of the energy consumed by module execution $\mathcal{E}_{exec}$ and the energy consumed by data transmission $\mathcal{E}_{trans}$, as represented in (8).

$$\mathcal{E}_{robot} = \mathcal{E}_{exec} + \mathcal{E}_{trans} \tag{8}$$

We also consider three situations mentioned in Section II.A. In the first case, the energy consumption can be calculated by (9), where the robot operates with a voltage of U, $C(m)$ represents the battery consumption by executing the module m, and $C(d)$ denotes the battery consumption by sending the data d from robot to cloud. Note that the unit of battery consumption is mAh, and $1mAh = 0.001A{*}3600s = 3.6A{\cdot}s$. In the second case, the energy consumption only comes from data transmission, and the energy consumption can be calculated by (10). In the third case, we do not use cloud resources and all modules are executed on the mobile robot. Therefore, $\mathcal{E}_{trans} = 0$, and the energy consumption can be calculated by (11).

$$\mathcal{E}_{robot} = \sum_{m \in \mathcal{M}_R} 3.6 * (C(m) * U + C(d) * U) \tag{9}$$

$$\mathcal{E}_{robot} = 3.6 * C(d) * U \tag{10}$$

$$\mathcal{E}_{robot} = \sum_{m \in \mathcal{M}} 3.6 * C(m) * U \tag{11}$$

C. Problem Formulation

To achieve real-time instance segmentation on the mobile robot with the aid of cloud, the key problem to be tackled is how to offload the computation of the instance segmentation network into the cloud. With the above formulation, the computation offloading problem can be transformed into deciding which nodes in the directed acyclic graph to be executed on the mobile robot and which nodes to be executed on the cloud platform, with the goal of minimizing the time cost of the whole instance segmentation and the energy consumption on the mobile robot, as shown in (12).

$$\min_{M_R, M_C} \alpha t_{total} + \beta \mathcal{E}_{robot} \tag{12}$$

III. Real-Time Instance Segmentation Framework Based on Computation Offloading

In this section, a real-time instance segmentation framework is presented for low-cost mobile robots. The key idea of our framework is to offload part of computation into the cloud, whereby the mobile robot can utilize sufficient cloud resources to fulfill the instance segmentation task and achieve real-time performance. For this purpose, the instance segmentation network is first divided into multiple modules and modelled as the directed acyclic graph, as presented in Section II. Afterwards, the offloading point in the directed acyclic graph is determined according to both module execution time and data transmission time, where the front segment will be executed on the mobile robot and the back segment will be executed on the cloud platform. Due to the limited communication bandwidth, the data transmitted from robot to cloud is further compressed with quantization technique, so that the communication cost can be minimized. The details about the proposed computation offloading strategy will be introduced as follows with the example of the representative one-stage instance segmentation method, Yolact [2].

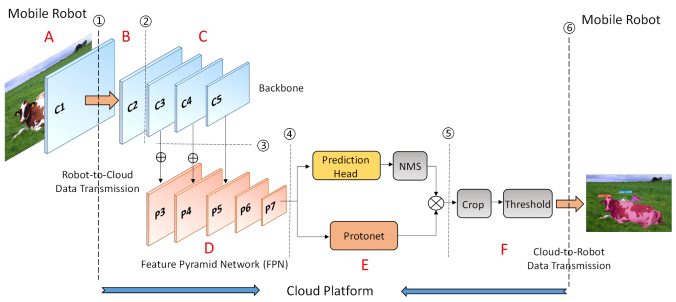

Fig. 2. The computation offloading strategy for Yolact.

As shown in Fig. 2, Yolact consists of a backbone network and a feature pyramid network (FPN), which are used to extract the image features, followed by two parallel subtasks, generating prototype masks (Protonet) and generating mask coefficients (Prediction Head), which are used to compute the instance masks, and the final detect component and a series of post-processing steps (Detect). Considering the network structure, Yolact can be divided into six sequential modules, including C1, C2, C3+C4+C5, FPN, Protonet+Prediction Head, and Detect, marked as A, B, C, D, E, and F in Fig. 2.

TABLE I
The time cost of each module on the mobile robot and cloud platform and the transmitted data size.

Modules	t_r(ms)	t_c(ms)	d(MB)
C1	627.1	0.2146	4.9
C2	1978.6	1.3616	19.5
C3+C4+C5	5564.2	5.8121	17.4
FPN	978.9024	0.8823	6.6
Proto+Pred_head	6313.7975	30.2746	11.7
Detect	217.7686	11.4515	2.2

After finishing the module division, the offloading point should be determined. As presented in equation (12), our goal is to minimize the time cost of the whole instance segmentation and the energy consumption on the mobile robot. We assume that the energy consumption depends on the time cost, and the time cost less, the energy consumption less. Therefore, the goal can be simplified into minimizing the total time. First, we can obtain the time cost of each module on both the mobile robot and cloud platform by experimental measurement, and compute the data size transmitted between adjacent modules. As shown in TABLE. I, t_r and t_c represent the time cost of each module on the mobile robot and the cloud platform, respectively, and d represents the output data size of modules. Then, the optimal offloading point can be computed by genetic algorithm, where different modules are encoded as different genes, and the execution time of module and the time caused by the communication between robot and cloud are represented as the allele value of gene. Finally, we choose the point between C1 and C2 as the offloading point for Yolact. Only C1 is executed on the mobile robot and the rest is executed on the cloud platform.

However, the transmitted data size is still large for limited communication bandwidth and will cause the time delay. To address this problem, we first compress the data with the quantization technique [8], and then send them to the cloud for subsequent processing. The quantization technique maps the data in the type of float32 into the data in the type of uint8. The transformation can be calculated by (13), where d and d_q refer to the data before quantization and after quantization, respectively. S represents the scale factor between the data before quantization and after quantization, and can be calculated by (14), where r_{max} and r_{min} are the maximum and minimum value of d, and q_{max} and q_{min} are the maximum and minimum value of d_q. Z represents the zero point of quantized data, and can be calculated by (15).

$$d_q = round(\frac{d}{S} + Z) \tag{13}$$

$$S = \frac{r_{max} - r_{min}}{q_{max} - q_{min}} \tag{14}$$

$$Z = round(q_{max} - \frac{r_{max}}{S}) \tag{15}$$

IV. Experiments and Discussions

A. Experiment Settings and Metrics

To validate the feasibility and efficiency of our framework, we applied the proposed framework to the one-stage instance segmentation network, Yolact, and the two-stage instance segmentation network, Mask R-CNN [1], respectively. The performance of networks was tested and reported under three situations:

1) Partly offloading: refers to offloading part of computation of network into the cloud using our proposed computation offloading strategy in Section III;

2) Entirely offloading: refers to offloading all computation of network into the cloud;

3) No offloading: refers to finishing all computation of network on the mobile robot.

The following two metrics were used to evalute the performance of networks:

1) Average Time Cost: refers to the inference time of the instance segmentation network on average per frame;

2) Average Energy Consumption: refers to the energy consumption on the mobile robot on average per frame;

In our experimental platform, the mobile robot platform adopted TurtleBot3 Waffle Pi with Raspberry Pi 4 Model B (4-core processor at 1.5GHz and 4GB memory), and the cloud platform was build with two Intel E5-2620V4 (8 cores @ 2.1GHz-3.0GHz) and 64GB memory, as well as one 2080Ti with 11G video memory. Note that the communication bandwidth in the experimental environments was about 20mbps.

B. Results and Discussions

First, we present our method's performance on average time cost compared with other two situations (i.e., entirely offloading and no offloading).

TABLE II demonstrates the results on Yolact. It can be found that our method (i.e., partly offloading) takes the least time about one second to finish the inference of a frame, while no offloading takes up to 15 seconds and entirely offloading takes four seconds.

In TABLE II, we also show the time cost of each module in Yolact, where C1, C2, C3, C4, and C5 are five blocks in the backbone network, FPN refers to the feature pyramid network, Proto+Pred_Head represents two parallel subtasks (i.e., generating prototype masks and predicting mask coefficients), and Detect represents the final detection stage and a series of post-processing steps. It can be found that the time cost of C2, C3+C4+C5, FPN, Proto+Pred_head, and Detect under partly offloading decreases greatly compared with no offloading. This is because these modules are offloaded into the cloud and the execution time on the cloud is far less than that on the mobile robot.

Otherwise, we also report the time cost of data transmission (i.e., Data Trans). Under entirely offloading, data transmission time accounts for more than the 95 percentage of the total time, although it takes very short time on computation. On the contrary, our method (i.e., partly offloading) makes a good trade-off between the computation time and data transmission time.

Similarly, TABLE III demonstrates the results on Mask R-CNN. C1, C2, C3, C4, and C5 also refer to five blocks in the backbone network, FPN represents the feature pyramid network, RPN represents the region proposal network, and ROI heads output the final detection results, including predicted classes, boxes, and masks. The results show that partly offloading takes the shorest time among three situations.

Second, we report our method's performance on energy consumption compared with other two situations (i.e., entirely offloading and no offloading). Fig. 3 and Fig. 4 illustrate the energy consumption on the mobile robot for Yolact and Mask R-CNN, respectively. Note that horizontal axis represents

TABLE II
THE AVERAGE TIME COST OF YOLACT UNDER PARTLY OFFLOADING, ENTIRELY OFFLOADING, AND NO OFFLOADING.

Modules	partly offloading (ms)	entirely offloading (ms)	no offloading (ms)
C1	627.1	0.2146	627.1
C2	1.3616	1.3616	1978.6
C3+C4+C5	5.8121	5.8121	5564.2
FPN	0.8823	0.8823	978.9024
Proto+Pred_head	30.2746	30.2746	6313.7975
Detect	11.4515	11.4515	217.7686
Data Trans	480	4040	0
Total	**1156.8821**	**4089.9967**	**15681.438**

TABLE III
THE AVERAGE TIME COST OF MASK R-CNN UNDER PARTLY OFFLOADING, ENTIRELY OFFLOADING, AND NO OFFLOADING.

Modules	partly offloading (ms)	entirely offloading (ms)	no offloading (ms)
C1	1750.0298	18.8074	1750.0298
C2+C3+C4+C5	18.5403	18.5403	19688.9774
FPN	5.4411	5.4411	10162.93
RPN	28.3043	28.3043	8721.5149
ROI heads	17.9994	17.9994	4869.8716
Data trans	1400	4040	0
Total	**3220.3149**	**4129.0925**	**45193.3237**

the number of experimental tests. The results show that our method (i.e., partly offloading) consumes much fewer energy on the mobile robot than no offloading. Note that although entirely offloading consumes the fewest energy on the mobile robot for instance segmentation tasks, but it needs more time cost.

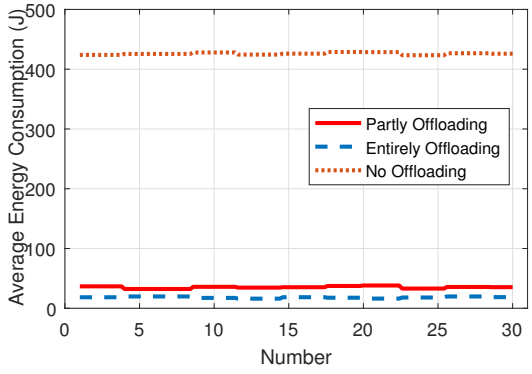

Fig. 3. The average energy consumption for Yolact on the mobile robot.

V. CONCLUSION AND FUTURE WORK

This paper proposes a real-time instance segmentation framework for low-cost mobile robot systems, which offloads part of computation of the instance segmentation network to the cloud, and leverages the computation and memory resources on the cloud platform to accelerate the network. The results demonstrate that our framework can greatly reduce

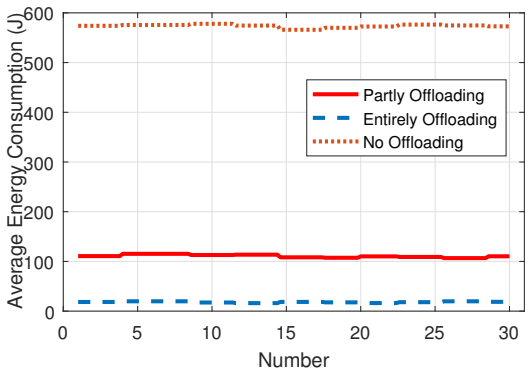

Fig. 4. The average energy consumption for Mask R-CNN on the mobile robot.

the time cost of the instance segmentation network on mobile robots, and achieve the performance of around one second per frame.

In the future, we will optimize both the speed and accuracy of our framework, and apply the instance segmentation methods to other robotic applications to facilitate the intelligent robot.

ACKNOWLEDGMENT

This work has been supported by the National Natural Science Foundation of China under Grant No. 61772068; the China Postdoctoral Science Foundation under Grant No. 2020M680352; the Guangdong Basic and Applied Basic Research Foundation under Grant No. 2020A1515110463; the Scientific and Technological Innovation Foundation of Shunde Graduate School, USTB, under Grant No. 2020BH011; and the Fundamental Research Funds for the Central Universities under Grant No. FRF-TP-20-063A1Z.

REFERENCES

[1] K. He, G. Gkioxari, P. Dollăr and R. Girshich, "Mask R-CNN," Proceedings of the IEEE International Coference on Computer Vision, 2017.
[2] D. Bolya, C. Zhou, F. Xiao and Y. J. Lee, "YOLACT:Real-time instance segmentation," Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
[3] D. Bolya, C. Zhou, F. Xiao and Y. J. Lee, "YOLACT++:Better real-time instance segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, in press.
[4] L. Deng, G. Li, S. Han, L. Shi and Y. Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," Proceedings of the IEEE, vol. 108, no. 4, pp. 485-532, 2020.
[5] Y. Guo, Z. Mi, Y. Yang and M. S. Obaidat, "An energy sensitive computation offloading strategy in cloud robotic network based on GA," IEEE Systems Journal, vol. 13, no. 3, pp. 3513-3523, 2019.
[6] Y. Xie, Y. Guo, Z. Mi, Y. Yang and M. S. Obaidat, "Loosely-coupled cloud robotic framework for QoS-driven resource allocation based web service composition," IEEE Systems Journal, vol. 14, no. 1, pp. 1245-1256, 2020.
[7] R. Liao, Y. Guo, Z. Mi, and Y. Yang, "Segmental deployment of neural network in cloud robotic system," 2018 IEEE 3rd International Conference on Cloud Computing and Internet of Things (CCIOT), 2018.
[8] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," Proceedings of the IEEE conference on computer vision and pattern recognition, 2018.