2018 IEEE Confs on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics

Real-time Object Recognition Based on NAO Humanoid Robot

Qianyuan Liu, Chenjin Zhang, *Member, IEEE*, Yong Song, and Bao Pang

***Abstract*—This paper focuses on the real-time object recognition based indoor humanoid robots like Nao robots. Improving the perceptive ability of service robot has always been a research hotspot. The breakthrough of computer vision technology represented by object recognition provides a broader idea for this purpose. We deployed a micro-cloud layer that connects the robot with the computer vision, thereby realized the concepts of RaaS (Robot as a service). In this paper, in order to make the Nao robot to detect objects faster. We present an architecture about real-time object recognition on Nao, and offload the task of control and data collection from robot to a PC. Next, the image data is transmitted over Ethernet to the workstation, which runs multiple parallel image processing services. These services are built with the current popular deep neural network by TensorFlow and running on a GPU GTX1080 Ti. In the micro-cloud layer, we designed a universal robotic visual task queue model, and a PC registers the task queue to the LAN. There are multiple workers in the LAN, and each worker is an independent service processer. Service processer obtains the task queue from the network and processes the queue, and then the processer puts the results back to the manager. The experimental results of the Nao robot in the simulation and real word show that our model and method are effective. The robot can recognize about 90 kinds of common objects, and each frame of image processing time is about 100 milliseconds.**

***Index Terms*—Object Recognition, Robot Vision Systems, Cloud Computing Security, Convolutional Neural Networks.**

I. Introduction

As people expect more and more humanoid robots into the daily life, humanoid robots are given much higher cognitive abilities from environment is becoming increasingly indispensable. Many robots such as Service Robots, Assistive Robots and Education Robots have been given a lot of abilities based on computer vision technologies through the great effort of many researchers. Martinez *et al.* [1] proposed a vision system about object detection and recognition for assistive robots to process visual data in real time, which combining color, motion, and shape cues in a probabilistic manner. D. Nyga *et al.* [2] presented a novel knowledge-driven approach in robot perception framework called ROBOSHERLOCK to transform phrases stated in natural language, particularly identify objects that the robot has never encountered before. The results of the experiments indicated their robot could learn objection descriptions from the web in nature language. L. Y. Ku *et al.* [3] showed a visuomotor system based on hierarchical CNN features interacts with the environment and memorizes the consequences of actions, in order to predicting the consequences of actions. The experiments demonstrated the robot was able to forecast the sequence of actions learns from observations' RGB-D data directly and was capable to manipulate and capture the objects.

Some other applications of robot such as Visual-SLAM, Visual Navigation, Human–Robot Interaction, Medical Robot and so on. Thus, a precise and reliable object recognition real-time system is a paramount importance for intelligence of robot.

For instance, we investigated the task of recognizing or detecting daily objects real-time in domestic environments purely with ordinary monocular camera. Benefit from the rapid development of computer vision technology based on deep learning, large-scale and complex target recognition algorithms emerge in an endless stream with the support of high-performance GPU and big data. In recent years, computer gradually close to human performance in some areas such as the game of Go, image classification, Even more than humans. In 2015, David Silver *et al.* [4] designed the first computer program to defeat a Go professional and then AlphaGo earned the highest certification 9 Dan professional ranking. Two years later, a stronger upgraded version without human knowledge about Go: AlphaGo Zero that meaning Learning from scratch, which Zero wined 100 - 0 against the previous AlphaGo with 3 days' learning by play-self [5].

This paper is organized as follows: an abbreviated introduction to real-time object recognition, cloud robot based on deep learning and the micro cloud layer in Section 2. The architecture and implement of the robot vision system is described in Section 3 including platform, several frameworks and working process. Some experimental results of the real robot are shown in Section 4. At last, the conclusion and future work is presented in Section 5.

II. Related Work

A. Object Recognition

AlexNet [6] is the champion in the ImageNet Large Scale Visual Recognition Challenge in 2012 with the network achieved a top-5 error of 15.3%. Significantly, the AlexNet

Q. Liu, C. Zhang, Y. Song, and B. Pang are with the School of Mechanical, Electrical and Information Engineering, Shandong University at Weihai, Weihai 264209, China (e-mail: cjzhang@sdu.edu.cn).

978-1-5386-7975-3/18/$31.00 ©2018 IEEE
DOI 10.1109/Cybermatics_2018.2018.00131

contained only 8 layers and carried out more than 10.8 percentage points ahead of the runner up. Since then, object recognition based on convolutional neural network as feature extractor quickly caught people's attention and achieved a great breakthrough. For the purpose of optimizing region proposal, the popular methods of target detection develop from RCNN to Fast R-CNN, Faster R-CNN, even YOLO [6]. However, each method has its own advantages and disadvantages. YOLO (You Only Look Once) is the fastest among those approaches while the Faster RCNN is more accurate but slower. Thus, some other advanced methods would emerge, and according to the actual need to determine which method to be choose.

As we all know, robot is a comprehensive advanced technology platform, combined several advanced technologies in one. Weak computing resources could not match with a variety of sophisticated sensor mechanical institutions, especially image processing which has a requirement of extracting discriminative features from data representing an object and accurately classifying the object in a real indoor environment. Computing platform cannot solve the problem of low intelligence, low autonomy, high cost and other insufficient with relying on the carrying sensors. Suppose such a scene: a service robot is deployed in a house. When the host comes back from outside and feels thirsty, the robot is designed to manipulate a glass of water from drinking fountains for the host automatically. In this process, robot need to overcome several technical hurdles, for example, real-time object recognition, target location, dynamic grasp and so on. To resolve these issue, massive data needs to be processed quickly and accurately, especially graphics stream of data which required to have extremely powerful graphics capabilities to be competent. In the mentioned above, computer vision processing based on the deep learning techniques is always support by large-scale high-performance parallel computing with infrastructure such as GPU (Graphics Processing Unit), FPGA (Field Programmable Gate Array), ASIC (Application Specific Integrated Circuit), and even TPU (Tensor Processing Unit) has been used to boosting inference of AlphaGo. However, facing with such a huge amount of data processing tasks, robotic platform existing cannot deal with.

B. Cloud Robots

Therefore, the proposal of cloud robot provides a train of thought to solve these bottlenecks. Cloud robots make use of the powerful computing, storage and communication capabilities of the cloud platform to offload complex calculations such as data processing, planning and decision-making operations around the robot to the cloud platform, thereby greatly expanding the capabilities of the robot. For instance, typical manipulation problems of robots is that a robot is expected to grab a common object and prevents it from falling. This process must take several steps, including object recognition, object localization, grasp planning, and motion planning, particularly, real-time object recognition.

In 2010, Kuffner, J.J. at Carnegie Mellon proposed the concept of "cloud robots" [7]. The huge advantages and potential of cloud robots have caused the research boom of cloud robots in academia and engineering and implemented extensive applications in the fields of service, medical care, environment and exploration. The University of California, Berkeley used Willow Garage's PR2 robot and Google Target Recognition Engine based cloud platform to complete the task of target 3D capture by RGB-D data [8]. In 2014, the RoboEarth project was launched by Hunziker, D. *et al.* [9] at the Eindhoven University of Technology, where four robots work together in a simulated hospital environment to take care of patients. In the mean while these robots share information and learn from each other through interactions with cloud servers. Dogmus, Z. *et al.* [10] represented the REHABROBO-ONTO: a combination of rehabilitation robotics and physical medicine with considering simultaneously the advantages of ontologies and ontological reasoning. This rehabilitation robot based on cloud utilize a structured mothed to express reusable information in order to share with others. The principal advantage of cloud robots is the sharing of information and knowledge on a global scale by accessing the Internet. Beksi, W.J. *et al.* [11] proposed a Cloud-based Object Recognition Engine that is able to effectively execute data transfers in a robotic network, which is a distributed, modular, and scalable software architecture.

In the current area of the cloud robots, there is still not a universal software architecture for data transmission, exchange protocol. In fact, according to a broad definition in [12] as follows: any robot or automation system that relies on either data or code from a network to support its operation, a cloud robot system has a remote cloud and local robotic body with communication by the internet, e.g. Wi-Fi or Ethernet. Great progress has been made about the work of robot grasping, for instance, Dex-Net is a network of robust grasp 3D object planning with Multi-View Convolutional Neural Networks cloud-based. The remote cloud-based dataset currently including over 10,000 specific 3D object models and 2.5 million parallel-jaw grasps is utilized to develop strategies of robotic manipulation. In upgraded Dex-Net 3.0, J Mahler, J. et.al train a Grasp Quality Convolutional Neural Network (GQ-CNN) to classify suction grasp robustness from the robotic data of point clouds. The experiment of ABB YuMi is evaluated in a dataset of basic and typical objects, showing that Dex-Net 3.0 based GQ-CNN achieves 99% and 97% precision respectively [13-15].

It is because that Cloud-based Robotics exchange data and perform computation via networks and a range of privacy and security concerns has been raised. Especially, sensitive data generated by Cloud-connected robots and sensors, such as images from private household or corporate trade secrets. If all the robotic data and actions are controlled by the cloud platform, there will be a serious risk of hacker attacking. A hacker could take over a Cloud-connected robot, steal information and even use the robot to cause damage remotely.

C. Micro Cloud Layer

To protect privacy, sensitive data such as videos or images should be avoided from being directly and explicitly transmitted over the network. Tian, G. *et al.* [16] proposed a concept of “micro cloud layer”. For some special occasions,

sensitive and private knowledge is stored and processed in "micro cloud layer", which includes specific object oriented information like images of indoor environment, personal identifications of user and so on. Real-time object recognition based on deep learning includes two parts: training model and inference. The training model takes more time and more computing resources. While the inference requires more parallel computation and faster mathematical. According to their respective data processing characteristics, the former is suitable for remote cloud computing with large-scale data sets. The other is benefit for the micro cloud layer, because the inference is a personalized task. The service robots upload their image or audio data to micro cloud layer and download special results by the micro cloud layer.

In this paper, we propose a new method of combined deep learning and service robots. We put the tasks of inference in deep learning into the micro cloud layer and deploy the training model tasks of deep learning in the remote cloud. Then we implement several experiments with several popular real-time object recognition algorithms on Nao robot and a Nvidia GPU. The results show our mothed is feasible and effective.

III. Architecture And Implement

A. The Basic Specification of Nao Robot

NAO is a programmable, autonomous humanoid robot developed by Aldebaran Robotics company. As shown in Fig.1. We deploy in the experiment is his 5th version, it is 574 mm in height, 275mm in width, and weighs less than 5.4kg.

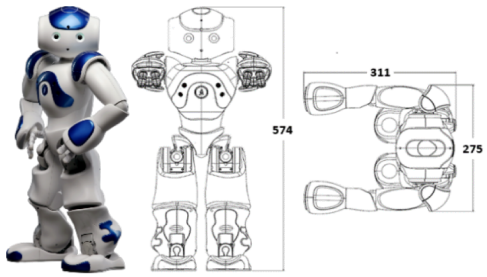

Fig. 1. The Introduction of the NAO's basic specification.

NAO is an interactive companion robot because of his 25 degrees of freedom for movement, two cameras, several touch sensors, four directional microphones, 2-axis gyro and ultrasonic sensor. Particularly, Nao could be controlled by C++ or Python language in a user-friendly programming environment. In addition, Nao also implement cute appearance and powerful function with robotic speakers and LEDs so that achieves high-level interaction with person, like autonomous move, animate dialogue.

B. The Vision of Nao Robot

There are two cameras used to NAO where one of the camera is located on the forehead and the other one is disguised the mouth.

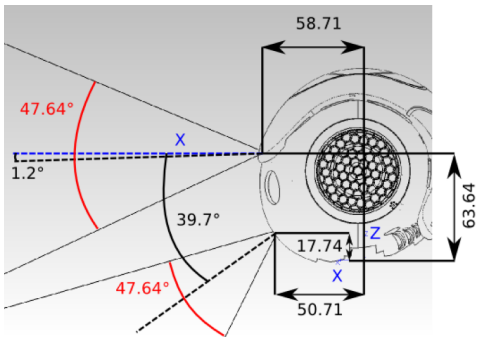

Fig. 2. Side view of the NAO cameras.

Nao robot has two identical video cameras are located in the forehead, which could provide a up to 1280x960 resolution at 30 frames per second and be used to identify objects in the visual field such as goals and balls.

The model of camera is MT9M114, as shown in Table I, there are some physical parameters that can be utilized to calibrate the cameras. The camera MT9M114 can only run internally from 5 to 30 fps.

TABLE I
THE PARAMETER OF CAMERA

Parameter Name	Description
Resolution	1.22 Mp
Camera output	1280*960@30fps
Data Format	(YUV[a]422 color space)
Field of view	72.6°DFOV (60.9°HFOV,47.6°VFOV)
Focus range	30cm ~ infinity
Focus type	Fixed focus
Pixel size	1.9μm*1.9μm
Dynamic range	70 dB
Signal/Noise ratio	37dB(max)
Optical format	1/6 inch
Active Pixels (HxV)	1288x968

Currently on ATOM CPU, requesting more than 5fps 1280x960 HD images remotely is bringing some frame drops.
[a] YUV colorspace is convenient as it is more powerful than RGB.

Moreover, Nao's camera can provide auto exposure algorithm and support many kinds of colorspaces. Specially, the cameras can be controlled to output pictures of different resolutions by programming keywords as shown in Table II.

TABLE II
SUPPORTED RESOLUTIONS ON THE NAO

Parameter ID Name	ID Value	Description[a]
AL::kQQQVGA	7	Image of 80*60px
AL::kQQVGA	0	Image of 160*120px
AL::kQVGA	1	Image of 320*240px
AL::kVGA	2	Image of 640*480px
AL::k4VGA	3	Image of 1280*960px

The camera MT9M114 can only run internally from QVGA to 4VGA, otherwise scale down is performed (without binning).
[a]The camera MT9M114 can only run internally from 5 to 30 fps.

The NAO is under controlled by Python program through Naoqi-Python-Interface in the remote PC. The camera transforms the surrounding environment information into a picture, which is then transmitted to a normal PC via Wi-Fi or Ethernet. A robot control and an image preprocessing algorithm program is running on the PC, and finally PC transmits the legal pictures to a workstation. The PC performs the preliminary filter processing on the picture, removes the noise and eliminates the ambiguous image. Then, running the real-time object recognition service process on the workstation could receive a request of detecting the object in a picture. In turn, the workstation feedbacks the result of detection containing every object's category and coordinate in the picture. The architecture about this application is shown in Fig.3.

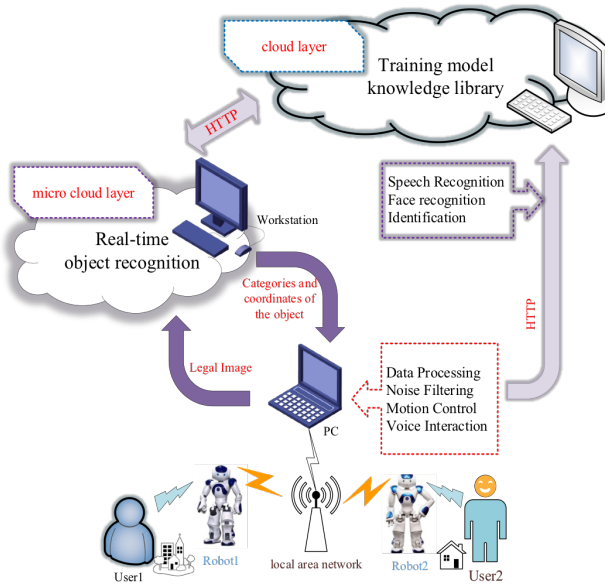

Fig. 3. The architecture of the robot vision system.

There is an important function what is to evaluate the quality picture on the PC side. Moreover, if an image is ambiguous or overexposed, then the blurry picture would be regulated by filter. In image deblurring methods, Gradient priors are widely used as they have been shown to be effective in suppressing artifacts. For an image x, the histogram of pixel intensity is different, we define

$$P_t(x) = ||x||_0 \quad (1)$$

where $||x||_0$ counts the number of nonzero values of x. Intuitive observation shows that the clear picture has sharper lines and feature than the blurry picture. Jinshan Pan et.al proposal use L_0-regularized prior, P_t (∇x), to model image gradients.

$$\mathrm{P(x)} = \sigma P_t(x) + P_t(\nabla x) \quad (2)$$

For the bad frame that could not be fixed, it will be abandoned.

C. *Preprocessing the Image of Robot Capturing*

So far, the preprocessed picture is sent to micro cloud layer, a workstation with a GPU. In the micro cloud layer, we deploy a distributed process on the PC and workstation for reducing time-varying network latency. Besides, the workstation is running worker multiprocessing for dealing the request from PC. There are two procession diagrams as shown in Fig.4 on the PC and Fig.5 on the workstation.

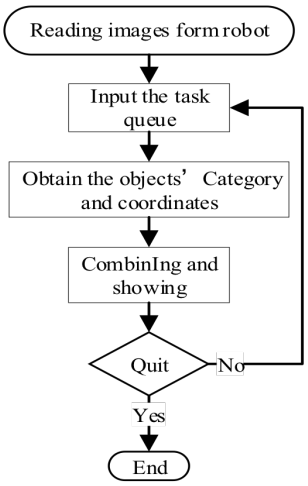

Fig. 4. Procession diagrams of PC.

For the PC, it looks like an information transfer station, because its connections to the robot and the workstation are both bi-directional. On the one hand, the PC takes the flow of sensor information from the robot and sends the control commands to the robot. On the other hand, it asks the workstation for image processing services and receives the

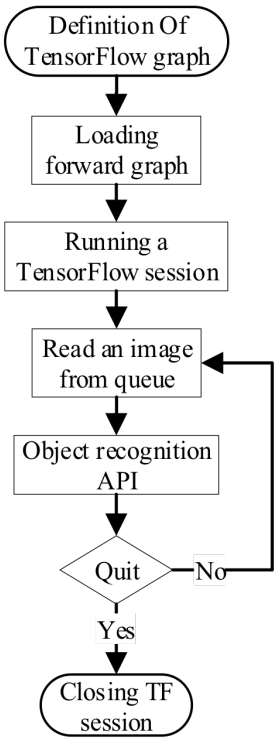

Fig. 5. Flow diagram of TensorFlow on the workstation.

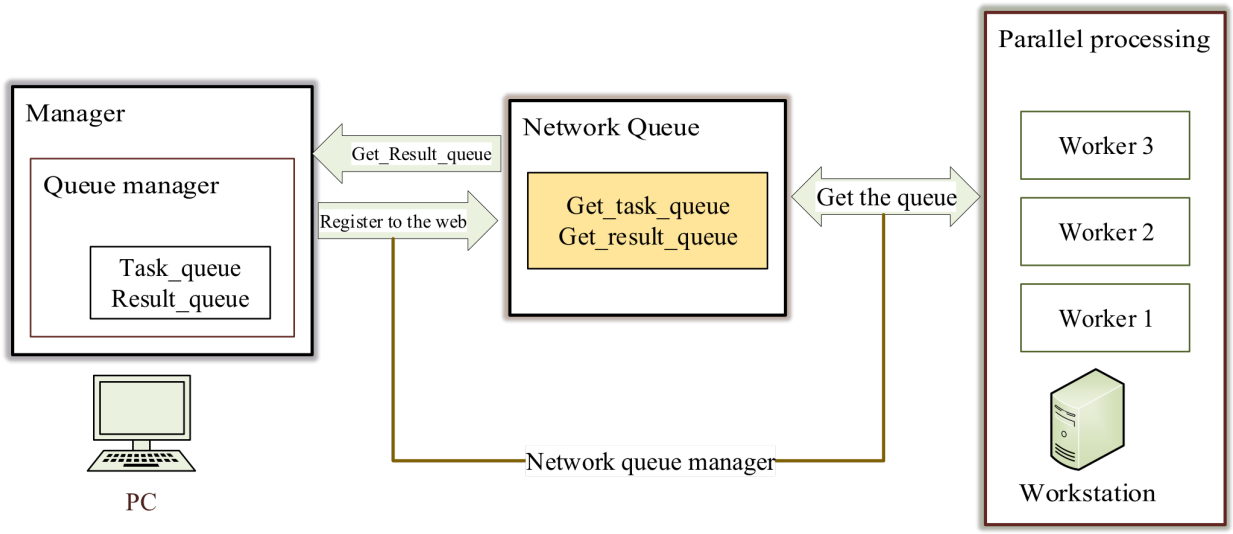

Fig. 6. The model of the task and result queue.

results of the image processing, then integrates these results and finally visualizes to the human through a GUI interface.

For the workstation, we use TensorFlow, a Google's high-performance computing framework. And its biggest feature is to define the graph and then use a variety of computational resources to calculate the graph caller session. TensorFlow uses a brief and clear dataflow graph to represent your complex and diverse computation. The most obvious advantage is that you can quickly define a novel algorithm like artificial neural network, with the attendant disadvantage of not being able to create a dynamic graph. Because the predefined calculation graph already contains complete logic and mathematical computational and cannot be changed during execution. However, the latest version TensorFlow1.6 already supports the definition of dynamic calculations.

D. TensorFlow Object Detection API

There are some remarks on frozen inference graphs as shown Table III. These pre-trained models' timings were performed using an Nvidia GeForce GTX TITAN X card and TensorFlow 1.4.0. These detection models are pre-trained on the COCO dataset.

TABLE III
DATASET TRAINED MODELS

Model name*	Speed (ms)	COCO[a] mAP[b]
SSD_MobileNet_v1_COCO	30	21
Faster_RCNN_ResNet50_COCO	89	30
RFCN_ResNet101_COCO	92	30
Faster_RCNN_Inception_ResNet_V2_atrous_COCO	620	37
Faster_RCNN_NAS	1833	43

*https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

[a]COCO is a large-scale object detection, segmentation, and captioning dataset.

[b]Mean average precision for a set of queries is the mean of the average precision scores for each query.

Our goal is to achieve the real-time of object detection, so we try our best to improve the speed of recognition, while keeping the accuracy at the same time. Therefore, we choose the model named SSD_MobileNet_V1_COCO (Single Shot MultiBox Detector). Inside the untarred directory, there are a graph proto, a checkpoint, a frozen graph proto with weights baked into the graph as constants to be used for out of the box inference and a config file which was used to generate the graph.

As shown in Fig.5, once the model is selected, the TensorFlow loads the corresponding calculation graph. Then, a TensorFlow session will be run. The task of this session is to read the images from the task queue constantly, and then to call the TensorFlow Object Detection API until it exits. The results returned by the API include the kind of object in the image and its confidence. Species tags below the set confidence threshold will be discarded.

IV. RESULTS AND DISCUSSIONS

In the experiment, we firstly verify the feasibility of the distributed process by simulation on the Webots. And then conduct experiments on several common scenarios with real robots to prove its validity by checking the categories and probabilities of objects.

As shown in Fig.6, The first step of the process is to get image from robot, then, the original picture is filtered to eliminate the noise, and it is determined whether the picture is ambiguous or not.

Put the clear picture into the Task_queue of Queue manager, mean while inquire the Result_queue. If there is any result in Result_queue, show the frame with the category and confidence of the object on the PC.

Repeat the above steps until exiting the application.

In the real world, it is obvious that wireless connection is far

less efficient than wired connection. Because Ethernet has much higher bandwidth than Wi-Fi. Therefore, improving the bandwidth of wireless communication also helps to improve the FPS of real-time target recognition system.

As shown in Fig. 7-9, the experimental results show that this system of robot vision can work effectively. In the simulation experiment, robot interact data with PC through the memory on the same machine, so the efficiency of object detection is the highest. In addition, wired Ethernet connections virtually eliminate the latency of network traffic. However, wireless connectivity is far less efficient than former. At last, experiments also illustrate that increasing the number of workers helps to increase FPS, but the number of workers is limited by the workstation's CPU.

Fig. 7. Simulation on Webots. Using software to simulate the real robot's physical state

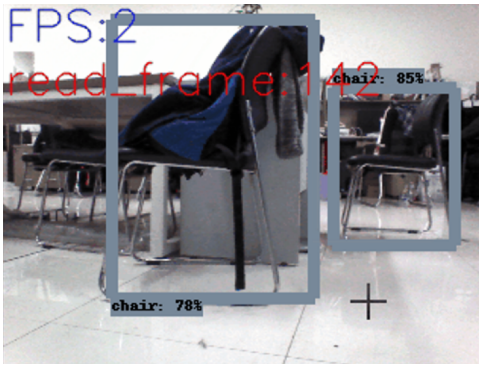

Fig. 8. Wireless connection of Nao robot and PC. This is recorded from the perspective of the real robot.

Fig. 9. Wired connection of Nao robot and PC. Using software to simulate the real robot's physical state

V. Conclusion

In this paper, we have introduced a novel architecture of real-time object recognition for our NAO robot in cloud computing. The robot can recognize about 90 objects in daily life, and the average processing time per frame of image is about 100 milliseconds. We deploy a deep neural network using TensorFlow on the workstation with a GPU GTX 1080 Ti. The robot vision system utilizes modular design, including neural network models, master-slave machines, image sources, robot distribution modules and so on. These modules are independent and freely changeable. In view of the diversity and complexity of robot vision tasks, we also propose an effective parallel queue model to reduce the delay of processing each frame of images uploaded by robots. Our next task is to connect several neural networks with different functions to add pedestrian detection and face detection, and finally realize the human-robot interaction based on recognizing facial expressions in a complex environment.

References

[1] E. Martinez-Martin and A. P. d. Pobil, "Object detection and recognition for assistive robots: Experimentation and implementation," *IEEE Robot. Autom. Mag.*, vol. 24, no. 3, pp. 123-138, 2017, 10.1109/MRA.2016.2615329.

[2] D. Nyga, M. Picklum, and M. Beetz, "What no robot has seen before — Probabilistic interpretation of natural-language object descriptions," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Singapore, 2017, pp. 4278-4285.

[3] L. Y. Ku, E. Learned-Miller, and R. Grupen, "An aspect representation for object manipulation based on convolutional neural networks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Singapore, 2017, pp. 794-800.

[4] D. Silver, A. Huang, C. J. Maddison *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016, 10.1038/nature16961.

[5] D. Silver, J. Schrittwieser, K. Simonyan *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354–359, 10/18/online 2017, 10.1038/nature24270.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, America, 2012, pp. 1097-1105.

[7] J. J. Kuffner and S. M. LaValle, "Space-filling trees: A new perspective on incremental search for motion planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, 2011, pp. 2199-2206.

[8] B. Kehoe, A. Matsukawa, S. Candido *et al.*, "Cloud-based robot grasping with the google object recognition engine," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Karlsruhe, Germany, 2013, pp. 4263-4270.

[9] D. Hunziker, M. Gajamohan, M. Waibel *et al.*, "Rapyuta: The RoboEarth Cloud Engine," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Karlsruhe, Germany, 2013, pp. 438-444.

[10] Z. Dogmus, E. Erdem, and V. Patoglu, "RehabRobo-Onto: Design, development and maintenance of a rehabilitation robotics ontology on the cloud," *Robot Comput Integr Manuf.*, vol. 33, pp. 100-109, Jun. 2015, org/10.1016/j.rcim.2014.08.010.

[11] W. J. Beksi, J. Spruth, and N. Papanikolopoulos, "CORE: A Cloud-based Object Recognition Engine for robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Hamburg, Germany, 2015, pp. 4512-4517.

[12] B. Kehoe, S. Patil, P. Abbeel *et al.*, "A survey of research on cloud robotics and automation," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 398-409, Jan. 2015, 10.1109/TASE.2014.2376492.

[13] J. Mahler, F. T. Pokorny, B. Hou *et al.*, "Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a Multi-Armed Bandit model with correlated rewards," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, 2016, pp. 1957-1964.

[14] J. Mahler, M. Matl, X. Liu *et al.*, "Dex-Net 3.0: Computing robust robot

suction grasp targets in point clouds using a new analytic model and deep learning," *arXiv preprint arXiv:1709.06670*, 2017.

[15] J. Mahler, J. Liang, S. Niyaz *et al.*, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[16] G. Tian, H. Chen, and F. Lu, "Cloud computing platform based on intelligent space for service robot," in *Proc. IEEE Int. Conf. on Information and Automation*, Lijiang, China, 2015, pp. 1562-1566.