Object Recognition System using Deep Learning with Depth Images for Service Robots

Yuma Yoshimoto[1] and Hakaru Tamukoh[2]
Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology
2-4 Hibikino, Wakamatsu, Kitakyushu 808-0196, Japan
[1] Email: yoshimoto.yuma276@mail.kyutech.jp
[2] Email: tamukoh@brain.kyutech.ac.jp

***Abstract*—In an aging society with fewer children, service robots are expected to play an increasingly important role in people's lives. To realize a future with service robots, a generic object recognition system is necessary to recognize a wide variety of objects with a high degree of accuracy. Therefore, this study employs deep convolutional neural networks for the generic object recognition system. To improve the accuracy of object recognition, both RGB images and depth images can be used effectively. In this paper, we propose a new architecture "Dual Stream - VGG16 (DS-VGG16)" for a deep convolutional neural network to train both the RGB images and depth images, and we also present a new training method for the proposed architecture. The experimental results indicate that the proposed architecture and training method are effective. Finally, we develop an object recognition system based on the proposed method that has an interface of robot operating system for integrating the system into service robots.**

***Index Terms*—Object Recognition, Depth Images, CNNs, Deep Learning, Service Robots,**

I. Introduction

In an aging society with fewer children, service robots are expected to play an increasingly important role in people's lives. The robots perform behaviors or tasks with a high degree of autonomy, such as house cleaning or delivering drinks at restaurants. To accomplish this, robots need a robust object recognition system that can recognize objects in various environments.

Since AlexNet won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, convolutional neural networks (CNNs) have become a standard for image recognition systems [1] [2] [3]. In the recent CNNs, VGG-16 is simple and extremely accurate [4]. The network achieved a TOP 5 error rate of 8.8% for the ImageNet Dataset (1000 categories of object recognition). However, if the robots fail to recognize objects in living spaces, the behavior of the robot may be affected, which can lead to human injury. Therefore, it is necessary to improve recognition accuracy as much as possible.

One method to improve accuracy is training RGB images with depth images. Several service robots have an RGB-D camera and can interpret RGB images and depth images. RGB images have information on appearance and texture. In contrast, depth images provide information about the shape of objects. The robots can obtain different information from RGB images and depth images. Eitel *et al.* reported that their network learns RGB images and depth images more accurately than CaffeNet, which trains RGB images only [5] [6]. However, their network is less accurate than the recently developed CNNs because the network has only a few layers.

This paper proposes a new network structure by combining VGG-16 with an effective method of training designed to interpret RGB images and depth images.

II. Related Work

A. Autonomous Service Robots

Service robots are robots that support people's lives in their home environments. Figure 1 shows the service robot "Exi@", which is researched and developed by our team [7]. This robot has various actuators and sensors to support human activities. The actuators include components such as an arm for gripping an object and a vehicle that allows the robot itself to move. The sensors include components such as an RGB-D camera for object recognition and a microphone for hearing human voices. In particular, RGB-D cameras can obtain both RGB images and depth images.

Depth images provide distance data from the RGB-D camera to objects surface. Figure 2 shows an RGB image and a depth image taken by the RGB-D camera. In depth images, objects near the camera (e.g., whiteboard, round table) are black, and objects further away from the camera (e.g., shelf) are white.

The depth images allow for the extraction of shape data independent from characteristics, such as object color or pattern. For example, Fig. 3 (a) and (c) are RGB images of balls in which a pattern is drawn. On the other hand, depth images, as shown in Fig. 3 (b) and (d), provide shape information, namely that both of the objects are spherical. Unlike RGB images, depth images are immutable to lighting and color changes, making shape information useful for recognizing objects with the same shape as being in the same class.

However, similar shape information can be obtained from an object of a different class if only depth images are used. In this case, RGB images work effectively to separate these classes. Therefore, it is necessary to train both RGB images and depth images.

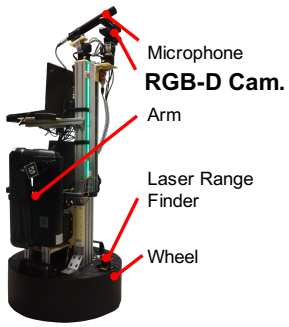
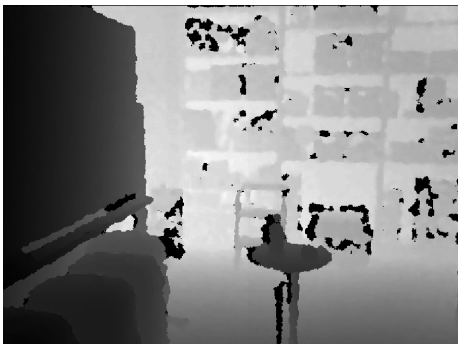


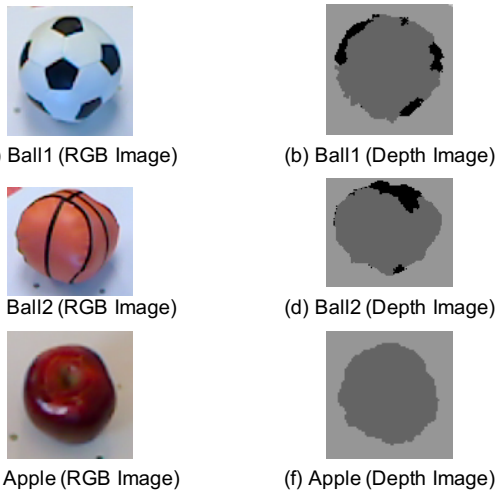
Fig. 1. Service Robot "Exi@"

(a) RGB Image



(b) Depth Image

Fig. 2. RGB-D Images



(a) Ball1 (RGB Image) (b) Ball1 (Depth Image)
(c) Ball2 (RGB Image) (d) Ball2 (Depth Image)
(e) Apple (RGB Image) (f) Apple (Depth Image)

Fig. 3. RGB-D Object Dataset

B. VGG-16

VGG-16 is one of the CNN models proposed by Karen Simonyan *et al.* at ILSVRC in 2014. This network achieved a TOP 5 error rate of 8.8% in the ImageNet recognition (1000 categories) at ILSVRC2012 and won 2nd place. In this network, 13 convolutional layers and 3 fully connected layers are connected in series. This configuration is simpler than GoogLeNet, which won this competition, because GoogLeNet consists of complicated branching and joining [8].

C. Eitel's network model

Eitel *et al.* proposed a network model of object recognition using RGB images and depth images for robot applications. Figure 4 shows the model. This uses "RGB Stream" for training RGB images and "Depth Stream" for training depth images. Each stream has the same CaffeNet model.

The training method of this network is described. First, preprocessing is performed and includes: (1) the expansion of the RGB images and depth images to match the input layer of the network (224 pixels $\times$ 224 pixels). The images are enlarged so that the long side of the image becomes 224 pixels while the aspect ratio is maintained; and (2) the colorization of the depth images. This method normalizes all the distance information for each pixel between 0 and 255. Then, these images are applied to a jet color map. In this process, depth images are converted into 3-channel color images.

Next, the network is trained using the preprocessed images and the following 2-steps: (1) RGB stream is pretrained using RGB images only. Furthermore, depth stream is pretrained using colorized depth images only; and (2) all network layers is trained using RGB images and depth images. By this method, this network model achieves an accuracy of 91.3% using the RGB-D Object Dataset [9].

D. Schwarz's method

Schwarz's method is one of an objects recognition system for robot application using RGB images and depth images [10]. Like Eitel's model, this model has two streams. However, in this model, Support Vector Machine (SVM) is connected to the end of the CNN. The SVM receives a feature map from the CNN and estimates the class or pose. In addition, the CNN uses the weight parameters already trained by ImageNet.

E. RGB-D Object Dataset

The "RGB-D Object Dataset" is one of the datasets for object recognition systems provided by the University of Washington. Figure 3 shows a part of the dataset, which includes 51 classes of objects in a typical home (e.g., apples, bananas, tissue boxes). In addition, the dataset consists of a total of 207,920 pairs of RGB images and depth images for all the classes. For these reasons, this dataset is an appropriate

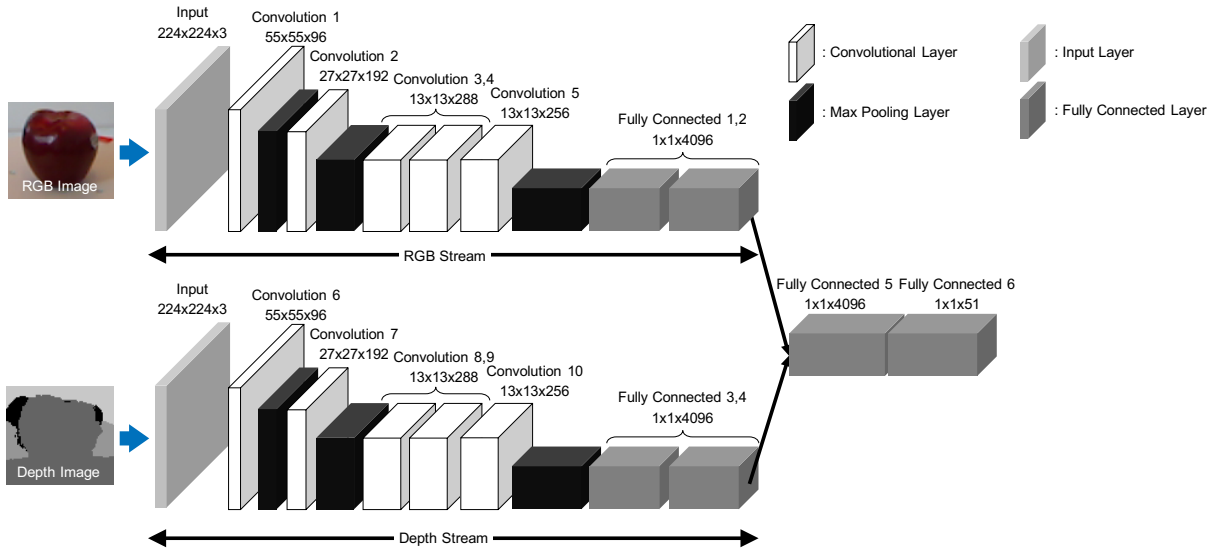


Fig. 4. Eitel's propose model

benchmark for general object recognition systems for service robots which are mounted RGB-D camera. Furthermore, in conventional methods such as Eitel's method, this dataset is used to verify accuracy. Therefore, in addition, we verify accuracy with the dataset in this research.

III. Proposed Method

In this paper, a new neural network "Dual Stream VGG-16 (DS-VGG16)" is proposed. This network improves accuracy using RGB images and depth images. In addition, we propose a training method for this network.

A. Network Design

Figure 5 shows the design of the proposed network. Similar to Eitel's method, this has "RGB Stream" for training RGB images and "Depth Stream" for training depth images. In addition, it uses a "Fully Connected 5-layer" structure that combines the output from both streams and estimates the class. Both RGB stream and depth stream have the same VGG-16 structure.

B. Training method

The network is trained using the following procedure; (1) preprocessing is performed. All of the RGB images and depth images are scaled to the size of the input layer. This process is the same in VGG-16; (2) each stream is pretrained using the scaled images. Each unit of convolutional 1 to 13 layers on RGB stream is used to record weight parameters. The depth stream is used to record weight parameters; and (3) fully connected layers are trained using both RGB images and depth images. The weight parameters of all convolutional layers are fixed pretraining values and are not affected by training. Fully connected layers 1 - 4 are initialized pretraining values, and fully connected layer 5, 6 are initialized random numbers.

C. Comparison to Eitel's method

Table I shows a comparison between the proposed method and Eitel's method. Compared to Eitel's method, the DS-VGG16 has more layers. Moreover, the proposed training method is simpler.

IV. Experiment

We used the RGB-D Object Dataset to evaluate the recognition accuracy of the DS-VGG16. Table II shows the experimental environment.

A. Setup

The RGB-D Object Dataset was used for evaluation, where the image of the dataset was randomly divided into 75% and 25% from each class. The network trained with 75% of the images in the dataset and performed the validation test with 25% of the images.

B. Train

First, we trained the DS-VGG16 in the all-layer training and the proposed training method. Then, we measured the accuracy of each method, and the results were compared. Finally, we compared the accuracy of the proposed method and previous methods.

1) All-layer Training: The proposed network used RGB images and depth images to train all layers simultaneously. The initial value of all unit weight parameters ware determined by random numbers. The network was trained 50 epochs and used a batch size of 25.

2) Proposed training method: The network was trained in the following procedure. First, VGG-16 was pretrained using the RGB images in the dataset. Initial weight parameters were random numbers. The network was trained 10 epochs and used a batch size of 25. After completion of training, the weight parameter of each unit was recorded in a file.

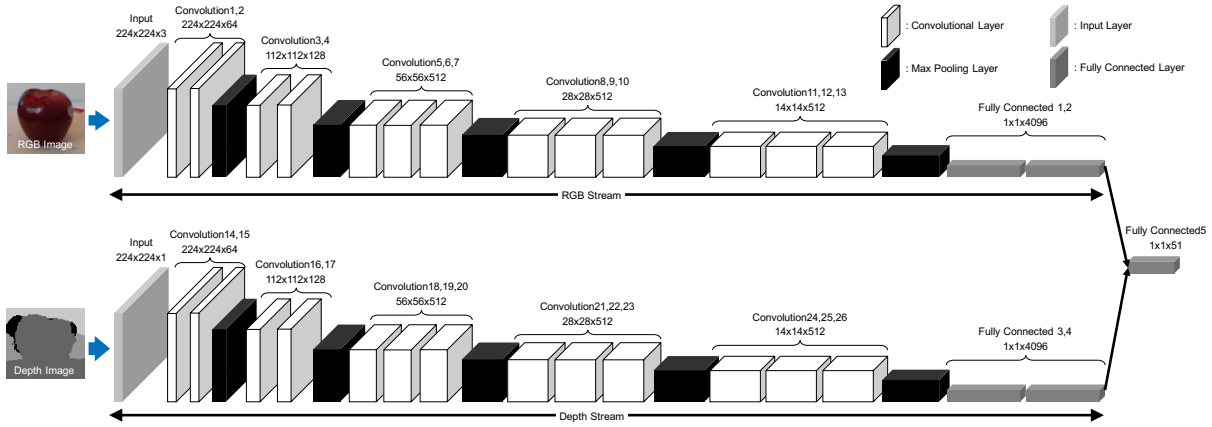


Fig. 5. DS-VGG16

TABLE I
COMPARE OF EITEL'S METHOD AND DS-VGG16

	Eitel's method [5]	DS-VGG16	Notes
Network design			
Built-in Network	CaffeNet	VGG-16	VGG-16 is built into DS-VGG16 because this network is high accuracy and simplicity.
Input Layer Size (RGB Stream)	224x224x3	224x224x3	Both networks are got 3ch input images.
Input Layer Size (Depth Stream)	224x224x3	224x224x1	DS-VGG16 does not use the convert to the JET color algorithm.
Number of Convolutional Layers	5 Layers x 2 Stream	14 Layers x 2 Stream	DS-VGG16 is based VGG-16, and there are more layers than Eitel's method.
Training method			
Pre-process for RGB images	Adjust aspect ratio	Fit to input layer	Pre-processing is same as VGG-16 because DS-VGG16 has VGG-16.
Pre-process for depth images	Adjust aspect ratio, convert to JET color images	Fit to input layer	Depth images are only processing change the sizeing process because we want training using raw data.
Pre-training (each stream)	All-layers training	All-layers training	In advance, we train each stream, networks can start the training.
Training	All-layers training	Only Fully Connected layer	In the proposed network, we learn only Fully Connected 1 to 5 layers without changing the filter weight obtained by Stream training.

TABLE II
EXPERIMENT ENVIRONMENT

CPU	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
Memory	331GB
GPU	NVIDIA Tesla P100
OS	Ubuntu14.04
Programming Language	Python 2.7.6
Framework	Chainer 1.17.1

Then, the DS-VGG16 was trained by copying the weight parameters of "Convolutional 1 - 13" and "Fully Connected 1, 2" on VGG-16 to RGB Stream built into DS-VGG16. Depth stream was pretrained and copied in a similar fashion. Finally, the DS-VGG16 was trained. All the convolutional layers had fixed weight parameters. The fully connected layers 1-4 were initialized with the copied weight parameters. Additionally, the fully connected 5-layer was initialized with random numbers. We trained DS-VGG16 50 epochs and a batch size of 25. After the completion of each epoch, we measure the validation accuracy.

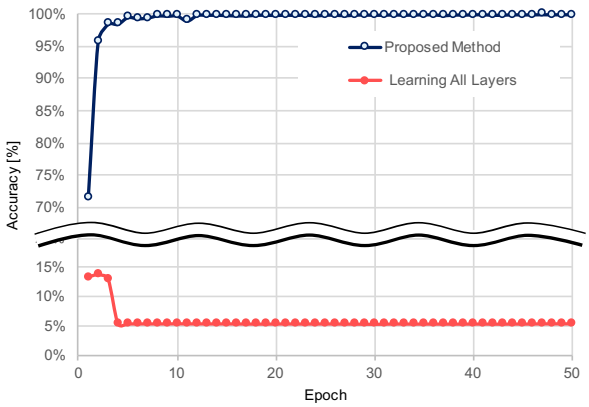


Fig. 6. Validation Accuracy

C. Results

1) Comparison of Training Methods: Figure 6 shows the results. In the full-layer training, the accuracy only shifted around 5%. On the other hand, in the proposed method, the accuracy reached 71% at 1 epoch, and the value also increased after that.

TABLE III
EXPERIMENT ON WASHINGTON RGB-D OBJECT DATASET

	RGB	RGB-D
Eitel et al. (2015) [5]	92.1%	94.1%
Schwarz et al. (2015) [10]	84.1%	91.3%
VGG-16 (2014) [4]	99.7%	–
DS-VGG16	–	99.9%

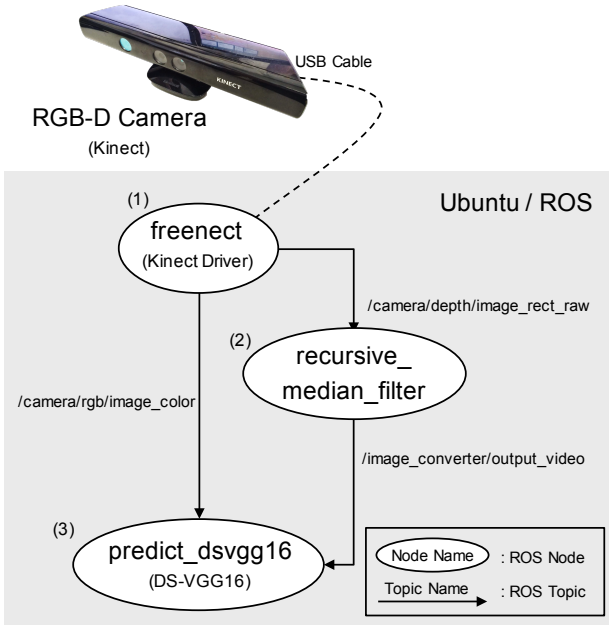


Fig. 7. ROS Implementation of DS-VGG16

2) Accuracy Validation of the Proposed Network: Table III shows the comparison results for the accuracy of DS-VGG16 and previous methods. The accuracy of the DS-VGG16 increased by 0.2 points compared to the VGG-16. In addition, the accuracy of the DS-VGG16 increased by 5.8 points compared to Eitel's method. Therefore, the DS-VGG16 and the proposed training method were effective for improving accuracy.

V. ROS IMPLEMENTATION

Robot Operating System (ROS) is a well-known middleware for robots [11]. We implement the DS-VGG16 with an ROS package and develop an object recognition system for service robots as mentioned in Sec. II-A. We verified it was able to classify images from the RGB-D camera using the DS-VGG16 ROS package. In this implementation, Microsoft Kinect was used as the RGB-D camera [12].

A. Software Diagram

Figure 7 shows our system, and each node is explained as follows:

1) freenect: A Freenect node is a Kinect driver [13]. This node takes the RGB images and depth images, then publishes them.

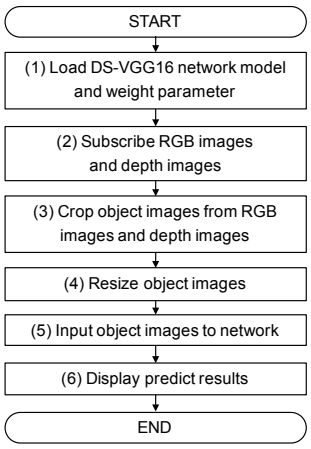


Fig. 8. Algorithm Flowchart

2) recursive_median_filter: Depth images have some missing values. This node fills up these missing depth values. First, this node subscribes the depth image from freenect. Next, the missing value in the depth images is filled with a recursive median filter. Finally, the processed depth images is published. The processing of this filter is the same as the K. Lai *et al.* method [9].

3) predict_dsvgg16: Figure 8 shows a flowchart for predict_dsvgg16 node. The processing flow of the node includes: (1) the node load DS-VGG16 network model and weight parameters. The weight parameters are those obtained in experiment 1 using the proposed training method; (2) RGB images and depth images are subscribed from freenect and recursive_median_filter nodes; (3) the object images are cropped from the RGB images and depth images. In this implementation, we set the cropping region in advance, place an object in the region; (4) RGB images and depth images are resized 224×224; (5) the DS-VGG16 network receives these images; and (6) the node displays the predict results.

B. Result

Some objects included in the RGB-D data set, such as bananas and tissue boxes, were taken by the Kinect camera. Figure 9 shows the RGB image and depth image inputted in DS-VGG16 network. Figure 10 shows the execution result of our system using the images as shown in Fig. 9. As a result, when input data similar to the images of the RGB-D Object Dataset were obtained, recognition results were correct. On the other hand, images which appeared differently from RGB-D Object Dataset (e.g., the images in which the entire objects were not shown, images with near or far to the objects) were obtained incorrectly result.

VI. CONCLUSION

In this study, we proposed the object recognition network "DS-VGG16" for service robots by deep neural networks using depth images, and the new network model included VGG-16. Moreover, we proposed the training method for DS-VGG16.

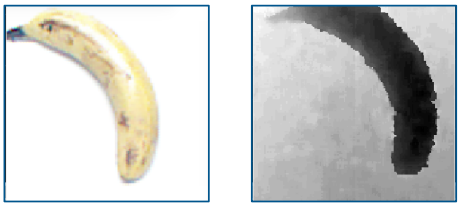


Fig. 9. Image Input to the Network on the predict_dsvgg16

```
predict start
predict end
------
2
class name is banana
```

Fig. 10. Execution Result

This method copied weight parameters to each stream from VGG-16, then training for all-layer. We used the RGB-D Object Dataset to verify the accuracy. The results indicated that the proposed method is effective, with accuracy exceeding that of conventional methods. Furthermore, we converted the DS-VGG16 to an ROS package and developed an object recognition system.

In the future, we will improve the DS-VGG16 by introducing more deeper or recent CNNs than VGG-16 to recognize images different from the trained dataset robustly. In addition, we will also introduce binary or quantized neural networks [14] [15] into the dual-stream structure to reduce calculation cost and memory usage then implement them into a field programmable gate array.

ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI grant number JP17H01798, JP17K20010. In addition, this paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, pp.1097–1105, 2012.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. F. Feiet, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol.115, no.3, pp.211–252, 2015.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," Proceedings of the IEEE, vol.86, no.11, pp.2278–2324, 1998.

[4] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556, 2014.

[5] A. Eitel, J.T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal Deep Learning for Robust RGB-D Object Recognition," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2015, pp.681–687, IEEE, 2015.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," Proceedings of the 22nd ACM International Conference on Multimedia, pp.675–678, ACM, 2014.

[7] S. Hori, Y. Ishida, Y. Kiyama, Y. Tanaka, Y. Kuroda, M. Hisano, Y. Imamura, T. Himaki, Y. Yoshimoto, Y. Aratani, K. Hashimoto, G. Iwamoto, H. Fujita, T. Morie, and H. Tamukoh, "Hibikino-Musashi@Home 2017 Team Description Paper," arXiv:1711.05457, 2017.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1–9, 2015.

[9] K. Lai, L. Bo, X. Ren, and D. Fox, "A Large-Scale Hierarchical Multi-View RGB-D Object Dataset," IEEE International Conference on Robotics and Automation (ICRA) 2011, pp.1817–1824, IEEE, 2011.

[10] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D Object Recognition and Pose Estimation based on Pre-trained Convolutional Neural Network Features," IEEE International Conference on Robotics and Automation (ICRA) 2015, pp.1329–1335, IEEE, 2015.

[11] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler and A. Ng, "ROS: an open-source Robot Operating System," ICRA workshop on open sourcesoftware, 2009.

[12] (2018, Jul.) Kinect for Windows. [Online]. Available: https://developer.microsoft.com/ja-jp/windows/kinect

[13] (2018, Jul.) freenect_launch. [Online]. Available: http://wiki.ros.org/freenect_launch

[14] M. Courbariaux, I. Hubara, D. Soudry, R. El Yani, and Y. Bengio, "Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or -1," arXiv:1602.02830v3, 2016.

[15] Y. Aratani, Y. Yoeng Jye, A. Suzuki, D. Shuto, T. Morie, and H. Tamukoh, "Multi-Valued Quantization Neural Networks toward Hardware Implementation," IEEE International Conference on Artificial Life And Robotics (ICAROB) 2017, pp.58, IEEE, 2017.