Distinction of Edible and Inedible Harvests Using a Fine-Tuning-Based Deep Learning System

Shinji Kawakura
Laboratory for Future Interdisciplinary Research in Science and Technology, Tokyo Institute of Technology, Yokohama, Japan
Email: s.kawakura@gmail.com,

Ryosuke Shibasaki
Center for Spatial Information Science, The University of Tokyo, Meguro, Japan
Email: shiba@csis.u-tokyo.ac.jp

***Abstract*—Effectively detecting and removing inedible harvests before or after harvesting is important for many agri-workers. Recent studies have suggested diverse measures, including various robot arm-based machines for harvesting vegetables and pulling up weeds, using camera systems to detect relevant coordinates. Although some of these systems have included monitoring and identification tools for edible and inedible targets, their accuracy has not been sufficient for use. Thus, further improvements have incorporated computing into the process based on human feelings and commonsense-based thinking, which considers up-to-date technologies and determines how solutions reflect the experience of traditional agri-workers. Our focus is on Japanese small- to middle-sized farms. Thus, we developed a fine-tuning (transfer-learning)-based deep learning system that gathers field pictures and performs static visual data analyses using artificial intelligence (AI)-based computing. In this study, pictures included kiwi fruits, eggplants, and mini tomatoes in outdoor farmlands. We focused on several program-based applications with deep learning-based systems using several hidden layers. To align with this year's technical trends, the data is presented concerning two patterns with different target layers: (1) all bonding layers with a revised pattern, and (2) some convolution layers with a visual geometry group (VGG) 16 and picture classifier created by convolutional neural network (CNN) revised pattern. Our results confirmed the utility of the fine-tuning methodologies, thus supporting other similar analyses in different academic research fields. In future, these results could assist the development of automatic agricultural harvesting systems and other high-tech agri-systems.**

***Index Terms*—picture classification, deep learning, fine-tuning, Keras, Theano**

I. Introduction

In recent years, agricultural researchers and workers (agri workers) have developed several automatic and mechanical techniques to improve the utility of harvesting robot-systems by enabling them to search for the color and size of vegetables and fruits based on visual data [1]-[4].

Manuscript received June 17, 2019; revised November 11, 2019.

These achievements in academic and business fields have already reached sufficient levels to utilize them in outer fields and inner farmlands. Additionally, researchers in the field of agricultural informatics and robotics have proposed various promising methods for improving these tasks. Existing visual analysis methods have focused mainly on vegetables, fruits, weeds, and farmers, including robotic farming systems in many ways [5]-[15], and other various targets [16], [17].

However, past studies and systems have been insufficient for fine-tuning based methodologies; that is why new technologies continue to be developed. And, in this study, we aims to develop a visual data analysis system by deep-learning, not based on open huge image datasets on the Internet [17], [18], but using original pictures, which connects to our program using libraries, external files and programs.

II. Materials and Methods

A. Field

We focused on Japanese traditional small- to middle-sized, non-trimmed outdoor farms to address requests from real farmers after our real hearing by oral.

B. Target

In this study, we used original pictures that we captured and aggregated in nonspecific outdoor farmlands. That is, we did not use available open picture datasets (e.g., sets in ImageNet 2012). First, we captured

1) kiwi fruits (kiwi), n = 162 (training data = 81, validation data = 81);

2) eggplants, n = 46 (training data = 23, validation data = 23); and

3) mini tomatoes, n = 64 (training data = 32, validation data = 32).

As shown in Fig. 1 and later in Table I, these amounts and weights were standardized video-analytically. Prior to the data collection, we consulted with agri-managers and workers because of the difficulties in handling dozens' sample numbers in the farmlands. Sets of square pictures of the targets (these pictures were parts of the

doi: 10.18178/joaat.6.4.236-240

data collection) were judged "edible" or "inedible" by experienced agri-workers (n = 3, their careers were over 20 years).

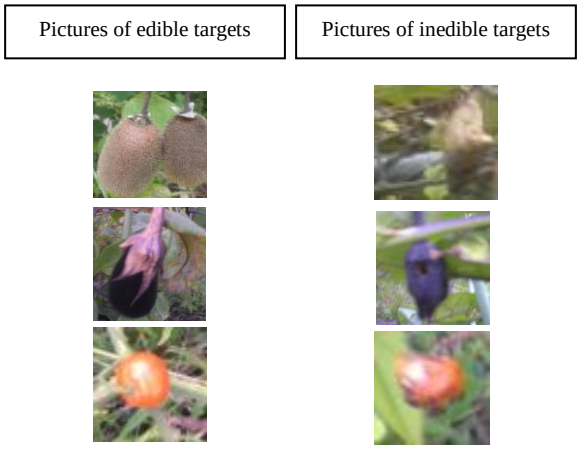

Figure 1. Example sets of square pictures of the targets judged by experienced agri-workers.

C. Analysis

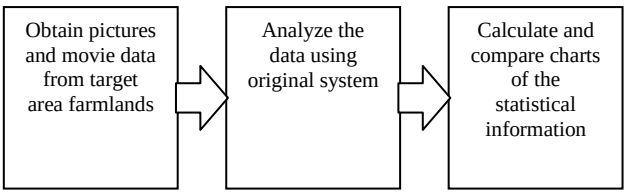

Figure 2. Flow of the experiment.

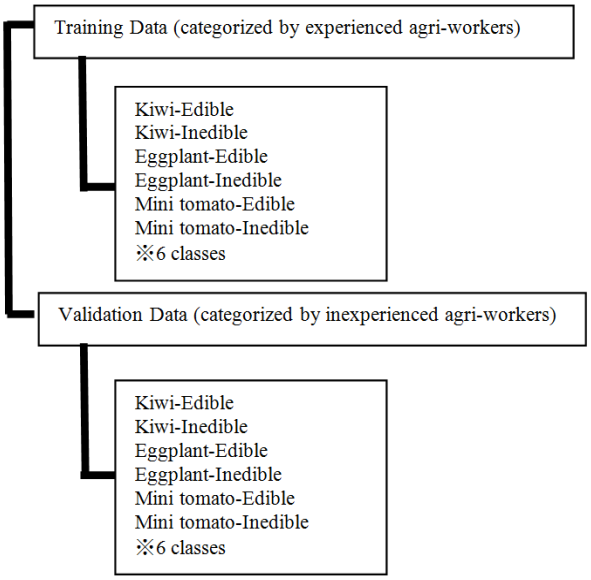

Figure 3. Classification of the pictures.

This study selected an artificial intelligence (AI)-based deep learning method, however did not use any open datasets (e.g., databases in ImageNet) for the target or training data. In light of current academic trends and past results, our methodology is adequate in the agricultural informatics field. Fig. 2 shows the flow of the experiment, which comprised (1) obtaining pictures and movie data from the target area farmlands, (2) analyzing the data using our programs (the adequacies of functions have been confirmed before), and (3) calculating and comparing charts of the statistical information. In future, we will present the results to agricultural system developers, agri-workers, and agri-managers. We set six picture classes: (1) Kiwi-Edible, (2) Kiwi-Inedible, (3) Eggplant-Edible, (4) Eggplant-Inedible, (5) Mini tomato-Edible, and (6) Mini tomato-Inedible, for Training Data (categorized by experienced agri-workers) and Validation Data (categorized by inexperienced agri-workers categorized) (Fig. 3).

Considering similar past trials, we uniformed the picture sizes to 224×224 pixels [2]-[15]. However, the complexity of the Japanese traditional, non-trimmed farmlands made it difficult to take measurements, and differences in the responses between individuals impeded understanding of the data.

In this study, aiming for increased accuracy, we performed a layer-oriented deep learning-based analysis. We used the latest Chainer framework and various peripheral programs (e.g., Anaconda), libraries, and packages.

In recent years, diverse gradient methods have been proposed for deep learning of picture data (Stochastic Gradient Descent (SGD), Momentum SGD (MSGD), AdaGrad, RMSprop, AdaDelta, Adam, etc.). We chose SGD as the optimizer for the system because of its effectiveness against redundancy concerning executions using the training data. We used one of the most common parameter value sets (*lr* = 0.0001, *momentum* = 0.9).

For the SGD's and MSGD's logics, we respectively iterate the w^t value in equations (1) and (2) as follows:

$$w^{t+1} = w^t - \eta \nabla f_n(w^t) = w^t - \eta (\partial E(w^t) / \partial w^t) \quad (1)$$

$$w^{t+1} = w^t - \eta \nabla f_n(w^t) + \alpha \Delta w^t = w^t - \eta (\partial E(w^t) / \partial w^t) + \alpha \Delta w^t \quad (2)$$

where E is the error function, η is the learning function, and α is the parameter of the inertia term. In standard gradient methods, we can solve common problems that the solved data are likely to set to the local optimum by randomly selecting samples with updating w^t values. This has the advantage of quickly learning the redundancy of the training data.

However, we must set the learning rate (coefficient) η arbitrarily, and we cannot change the settled η through the whole sequential process of error(s) minimization (for Chainer, η = 0.01 in default.). Thus, there are difficulties in choosing the most appropriate parameters according to the type of machine-learning.

For the process, particularly the fine-tuning of agri-pictures, we use the functions of Numpy, Blob, etc., and attempted to achieve a highly precise distinction rate using fewer pictures (presented in Table I) than are generally used in these (standard) technical fields (generally, over hundreds of pictures).

Table I shows a set of items used for the Chainer framework-based analyses, which comprised multiple

layers. As shown in Fig. 4, the sequential processing was programmed in Python, and the ratios of the areas between the main targets and the whole pictures were calculated using accuracy-assessed original programs. For system development, we used Python 3 to code the main program systems, and Theano as the library for the machine learning in the background of Keras 2.0. Theano and TensorFlow sat behind Keras 2.0, which is a neural-network library written in Python that we used to write the sample code. Additionally, we selected the visual geometry group (VGG) 16-model (known as VGG16) or picture classifier created by convolutional neural network (CNN); these are a commonly used, valid convolutional neural-network model.

[Pre-processing]
- Declaration of system's various path
- Setting parameters concerning Keras, and the backend running program Theano
- Importing VGG 16 (or CNN), and Image Data Generator
- Declaration of the pictures' category numbers, images' size, and batches' size

[Main function]
- Setting the pictures' size, and these numbers into "inout_tensor"
- Setting the base_model as VGG 16 (or CNN), and the parameters
- Setting other variables
- Compiling the layer model, and other small executions
- Outputting analyzed data sumarry

[Set TrainingDataGenerator as ImageDataGenerator]
- Setting variables
- Executing the setting TrainingDataGenerator as ImageDataGenerator

[Set TestingDataGenerator as ImageDataGenerator]
- Setting variables
- Executing the setting TestingDataGenerator as ImageDataGenerator

[Set items of TrainingDataGenerator]
- Setting variables

Figure 4. Steps of the main program for the trials.

After fine-tuning the model and executing machine learning, a user will be able to simply and quickly categorize objects concerning 1,000 category models, without requiring default installed pictures.

In the case where past pictures are used for learning and these are quite different to current trials, these analyzed characteristic points and values cannot be used directly. A user typically needs considerable training data and long computational time for machine learning. However, there are various patterns of fine-tuning, and a user may consider how to change the layers of the machine learning. That is, a user can freeze (stabilize) arbitrary layers; the benefit is mainly the flexible controlling of the speed, accuracy, and other characteristics of the analyses.

Considering this study as the first in a series, we compared the following methods: (1) changing all bonding layers (layers' weight) and freezing other layers; and (2) executing learnings to change weights of some bonding layers (Fig. 5 and Fig. 6). For Fig. 5, we used the aforementioned classified captured pictures in the respectively named data folders presented in Fig. 3. We did not renew the VGG16 layer, but executed the learning for the attached new layers.

Layer	Layer's Type	Output Shape
input_1	Input Layer	None, 224, 224, 3
block1_conv1	Conv2D	None, 224, 224, 64
block1_conv2	Conv2D	None, 224, 224, 64
block1_pool	MaxPoolomg2D	None, 112, 112, 64
block2_conv1	Conv2D	None, 112, 112, 128
block2_conv2	Conv2D	None, 112, 112, 128
block2_pool	MaxPoolomg2D	None, 56, 56, 128
block3_conv1	Conv2D	None, 56, 56, 256
block3_conv2	Conv2D	None, 56, 56, 256
block3_conv3	Conv2D	None, 56, 56, 256
block3_pool	MaxPoolomg2D	None, 56, 56, 256
block4_conv1	Conv2D	None, 28, 28, 512
block4_conv2	Conv2D	None, 28, 28, 512
block4_conv3	Conv2D	None, 28, 28, 512
block4_pool	MaxPoolomg2D	None, 14, 14, 512
block5_conv1	Conv2D	None, 14, 14, 512
block5_conv2	Conv2D	None, 14, 14, 512
block5_conv3	Conv2D	None, 14, 14, 512
block5_pool	MaxPoolomg2D	None, 7, 7, 512
global_average_pooling2d_1	-	None, 512
dense_1	**Dense**	**None, 1024**
dense_2	**Dense**	**None, 3**

Figure 5. The layers in the first method. (Changing all bonding layers).
The bold-enclosed area contains target layers for the learning.

Layer	Layer's Type	Output Shape
input_1	Input Layer	None, 224, 224, 3
block1_conv1	Conv2D	None, 224, 224, 64
block1_conv2	Conv2D	None, 224, 224, 64
block1_pool	MaxPoolomg2D	None, 112, 112, 64
block2_conv1	Conv2D	None, 112, 112, 128
block2_conv2	Conv2D	None, 112, 112, 128
block2_pool	MaxPoolomg2D	None, 56, 56, 128
block3_conv1	Conv2D	None, 56, 56, 256
ock3_conv2	Conv2D	None, 56, 56, 256
block3_conv3	Conv2D	None, 56, 56, 256
block3_pool	MaxPoolomg2D	None, 56, 56, 256
block4_conv1	Conv2D	None, 28, 28, 512
block4_conv2	Conv2D	None, 28, 28, 512
block4_conv3	Conv2D	None, 28, 28, 512
block4_pool	MaxPoolomg2D	None, 14, 14, 512
block5_conv1	Conv2D	None, 14, 14, 512
block5_conv2	Conv2D	None, 14, 14, 512
block5_conv3	Conv2D	None, 14, 14, 512
block5_pool	MaxPoolomg2D	None, 7, 7, 512
global_average_pooling2d_1	-	None, 512
dense_1	Dense	None, 1024
dense_2	Dense	None, 3

Figure 6. The layers in the second method. (Changing weights of some bonding layers based on machine learning). # The bold-enclosed area contains target layers for the learning.

Next, as presented in Fig. 6, we changed the area to be targeted for deep learning in the aforementioned layer model from "dense_1 to dense_2" into "block5_conv1 to dense_2." As presented above, we performed fine-tuning, and changed only limited layers to support computational speed using 272 (81 (pieces) × 2 (sets), 23 × 2, 32 × 2) pictures, as presented in Table I, for the machine-leaning. The recognition accuracy was then calculated. Through the process, we contrasted the results for the following two cases since the function is used in diverse cases across scientific fields: (1) the case to utilize the existing ImageDataGenerator and (2) the case for turning off the ImageDataGenerator.

III. Results

Fig. 7 and Fig. 8 illustrate these training accuracy and validation accuracy for the cases ImageDataGenerator OFF (Fig. 7) or ON (Fig. 8) concerning "Kiwi-Edible". The provided are the closest graphs to medium graph-line data of them. Table I presents the statistical results considering past studies utilizing the Chainer framework [12], [18]. The items in the rows of "Validation Accuracy" are average values of the calculation time. The case in Fig. 6 (changing some bonding layers) needed about eight times more calculation time, however, had a higher average accuracy than the case in Fig. 5 (changing all bonding layers).

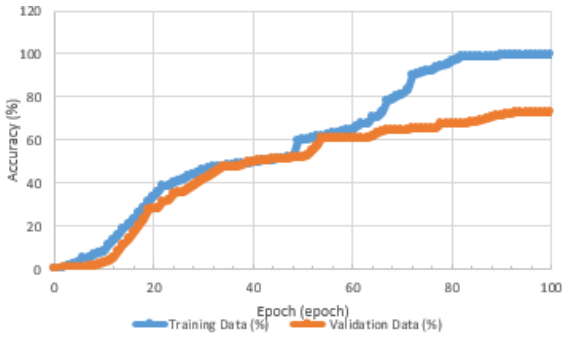

Figure 7. Training accuracy and validation accuracy in the case ImageDataGenerator OFF for Kiwi-Edible, Keras, and VGG16.

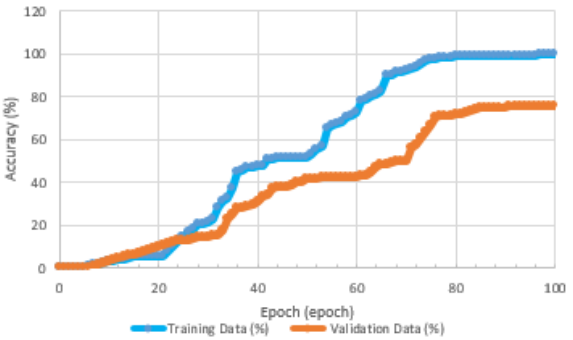

Figure 8. Training accuracy and validation accuracy in case ImageDataGenerator ON concerning Kiwi-Edible, Keras, and VGG16.

TABLE I. Validation Accuracy from 13 Trials

Model (Case)	Kiwi-Edible				Eggplant-Edible				Mini tomato-Edible			
	Keras and CNN		Keras and VGG16		Keras and CNN		Keras and VGG16		Keras and CNN		Keras and VGG16	
Number of Picture Data as (1) Training data and (2) Validation data	(1) 81, (2) 81				(1) 23, (2) 23				(1) 32, (2) 32			
Epoch	100	100	100	100	100	100	100	100	100	100	100	100
Fine-Tuning	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON
Image Data Generator	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON
Validation Accuracy (%)	61.1	63.9	72.5	75.9	55.5	57.9	58.1	59.9	68.2	72.0	75.4	76.7

IV. Discussion

Table I presents the numerical features of the cases of the kiwi, eggplant, and mini-tomato datasets.

In Fig. 7 and Fig. 8, the blue lines show the training case accuracy, and the red lines show the validation case accuracy. Comparing these graphs, the ImageDataGenerator ON case was slower to learn, however, ultimately had higher accuracy than

ImageDataGenerator OFF. In this study, we could not obtain statistically sufficient volumes of picture data, so the ideal graph-lines should increase further and quicker. For Fig. 7 and Fig. 8, if we increased the amount of data, the orange line could reach a higher level, indicating greater accuracy.

For the data in Table I, we observed the limitation concerning the range of all values from 55.5% to 76.7%. For three harvests, the lowest-accuracy patterns were "Fine-tuning = OFF" and "ImageDataGenerator = OFF," and the highest-accuracy patterns were "Fine-tuning = ON" and "ImageDataGenerator = ON." For the "Eggplant-Edible" data, four numerical data concerning "Validation Accuracy" (55.5% – 59.9%) were the lowest, perhaps because the eggplant has the darkest (non- vivid) colors. By contrast, the "Mini tomato-Edible" data showed the highest results.

In this phase, it was difficult to determine whether the sets of tools are suitable for judging whether actual agricultural items are edible or inedible. Specifically, relating to the system, it is difficult to comment concerning the combination of Keras, CNN, VGG16, and ImageDataGenerator from only these results. However, obvious differences are evident in Fig. 7 and Fig. 8.

V. Conclusion and Future Tasks

In this study, we constructed and demonstrated fine-tuning and deep learning-based visual data analysis for three harvests at agri-sites.

We analyzed our original captured picture files automatically considering various future practical usages, and presented various timeline and numerical data of classification accuracy, with changing conditions related to CNN, VGG16, ImageDataGenerator, etc.

Our future work will aim to provide further confirmation related to the variety of the detected targets and background conditions. Additionally, we may check the system durability, long-term performance, and other patterns or databases.

In the long term, these results could be used for automatic systems to help both indoor and outdoor farmlands improve their agri-work skills. We hope the aforementioned promising methodologies will be widely applied to real working sites to promote the recruitment of workers into agricultural fields.

References

[1] K. Ahlin, B. Joffe, A. P. Hu, G. McMurray, and N. Sadegh, "Autonomous leaf picking using deep learning and visual-servoing," *IFAC-Papers on Line*, vol. 49, no. 16, pp. 177-183, October 2016.

[2] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, and V. Kumar, "Counting apples and oranges with deep learning: A data-driven approach," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 781-788, December 2016.

[3] J. Dong, N. Sadegh, K. Ahlin, G. Rains, G. McMurray, B. Joffe, and B. Boots, "Robotics for spatially and temporally unstructured agricultural environments," in *Robotics and Mechatronics for Agriculture*, CRC Press, July 2017, pp. 58-82.

[4] N. Zhu, X. Liu, Z. Liu, K. Hu, Y. Wang, J. Tan, and Y. Guo, "Deep learning for smart agriculture: Concepts, tools, applications, and opportunities," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 4, pp. 32-44, July 2018.

[5] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, pp. 1-12, May 2016.

[6] Y. Sakai, T. Oda, M. Ikeda, and L. Barolli, "A vegetable category recognition system using deep neural network," in *Proc. 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS) IEEE*, July 2016, pp. 189-192.

[7] F. A. Guth, S. Ward, and K. P. McDonnell, "Autonomous disease detection in crops using deep learning," *Biosystems and Food Engineering Research Review*, vol. 22, pp. 1-209, May 2017.

[8] E. Cengil, A. Çınar, and E. Özbay, "Image classification with caffe deep learning framework," in *Proc. IEEE International Conference in Computer Science and Engineering*, October 2017, pp. 440-444.

[9] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299-315, April 2017.

[10] R. Wang, J. Zhang, W. Dong, J. Yu, C. J. Xie, R. Li, and H. Chen, "A crop pests image classification algorithm based on deep convolutional neural network," *Telkomnika*, vol. 15, no. 3, pp. 1239-1246, August 2017.

[11] Y. Huang, "The advancement of nature-inspired algorithms for agriculture," in *Proc. American Society of Agricultural and Biological Engineers (ASABE) Annual International Meeting*, July 2018, p. 1.

[12] M. Ikeda, Y. Sakai, T. Oda, and L. Barolli, "A vegetable category recognition system: A comparison study for Caffe and Chainer DNN frameworks," *Proceedings of Soft Computing*, pp. 1-8, April 2017.

[13] F. J. Rodríguez, A. García, P. J. Pardo, F. Chávez, and R. M. Luque-Baena, "Study and classification of plum varieties using image analysis and deep learning techniques," *Progress in Artificial Intelligence*, vol. 7, no. 2, pp. 119-127, January 2018.

[14] A. Patino-Saucedo, H. Rostro-Gonzalez, and J. Conradt, "Tropical fruits classification using an AlexNet-type convolutional neural network and image augmentation," in *Proc. International Conference on Neural Information*, Dec. 2018, pp. 371-379.

[15] F. Femling, A. Olsson, and F. Alonso-Fernandez, "Fruit and vegetable identification using machine learning for retail applications," in *Proc. 14th International Conference on Signal-Image Technology & Internet-Based Systems*, Jan. 2018.

[16] A. Dutta, J. M. Gitahi, P. Ghimire, and R. Mink, "Weed detection in close-range imagery of agricultural fields using neural networks," *Publikationen der DGPF, Band 27*, pp. 633-645, 2018.

[17] T. Zin, C. N. Phyo, P. Tin, H. Hama, and I. Kobayashi, "Image technology based cow identification system using deep learning," in *Proc. the International Multi Conference of Engineers and Computer Scientists*, March 2018, vol. 1, pp. 320-323.

[18] Y. Jia and E. Shelhamer. (September 2018). Blobs, layers, and nets: Anatomy of a Caffe model. Caffe Berkeleyvision. [Online]. Available: http://caffe.berkeleyvision.org

Shinji Kawakura was born in Toyama Pref., Japan on July 14, 1978. He received Ph.D. in Environmentology at University of Tokyo in 2015, Bunkyo-ku, Tokyo, Japan; B.A. in Control System Engineering at Tokyo Institute of Technology in 2003, Meguro-ku, Tokyo, Japan; M.A. in Human-Factor Engineering at Tokyo Institute of Technology in 2005, Meguro-ku, Tokyo, Japan.

His career includes Systems engineering, research for private companies. Development and verification of sensing systems for outdoor agricultural workers.

Dr. kawakura, Laboratory for Future Interdisciplinary Research in Science and Technology, Tokyo Institute of Technology, Yokohama, Japan. Committee member of ICEAE and ICBIP.

Ryosuke Shibasaki is with Department of Socio-Cultural and Socio-Physical Environmental Studies, The University of Tokyo/Kashiwa-shi, Chiba, Japan. Dr. in Engineering. Professor at the Center for Spatial Information Science, University of Tokyo.