# Grading Fruits and Vegetables Using RGB-D Images and Convolutional Neural Network

Toshiki Nishi
*Department of Control Engineering*
*Kyushu Institute of Technology*
Kitakyushu, Japan
nishi@kurolab.cntl.kyutech.ac.jp

Shuichi Kurogi
*Department of Control Engineering*
*Kyushu Institute of Technology*
Kitakyushu, Japan
kuro@cntl.kyutech.ac.jp

Kazuya Matsuo
*Department of Control Engineering*
*Kyushu Institute of Technology*
Kitakyushu, Japan
matsuo@cntl.kyutech.ac.jp

***Abstract*—This paper presents a method for grading fruits and vegetables by means of using RGB-D (RGB and depth) images and convolutional neural network (CNN). Here, we focus on grading according to the size of objects. First, the method transforms positions of pixels in RGB image so that the center of the object in 3D space is placed at the position equidistant from the focal point by means of using the corresponding depth image. Then, with the transformed RGB images involving equidistant objects, the method uses CNN for learning to classify the objects or fruits and vegetables in the images for grading according to the size, where the CNN is structured for achieving both size sensitivity for grading and shift invariance for reducing position error involved in images. By means of numerical experiments, we show the effectiveness and the analysis of the present method.**

***Index Terms*—grading fruits and vegetables according to size, RGB-D images, convolutional neural network, size sensitivity, shift invariance**

## I. Introduction

This paper presents a method for grading fruits and vegetables by means of using RGB-D (RGB and depth) images and convolutional neural network (CNN). Here, grading of fruits and vegetables, in general, is sorting them into different grades according to the size, shape, color, volume, and so on, and we focus on grading according to the size in this paper. There are various methods using artificial neural networks and image processing for classification and grading of fruits and agricultural products as reviewed in [1]–[3], where features such as size, shape, color, texture, etc. are used for grading. However, no research study on grading using RGB-D images and CNN is reviewed, while, recently, a number of object classification and/or recognition methods using RGB-D images and CNNs are studied [4]–[6]. Although they have shown state-of-the-art performance in classification and/or recognition tasks, one of the problems to use CNNs for grading objects according to size is that the ability of CNNs for size invariant classification may be ineffective. Furthermore, CNNs require a fixed size input and widely used image warping approach [5] will cause loss of size information.

By means of considering these aspects, we have constructed a method as follows; first, the method transforms positions of pixels in RGB image so that the center of the object in 3D space is placed at the position equidistant from the focal point by means of using the corresponding depth image. Then, with the transformed RGB images involving equidistant objects, the method uses CNN for learning to classify the objects for grading, where the CNN is structured for achieving both size sensitivity for grading and shift invariance for reducing position error involved in the images. We show the details of the method in II, experimental results and analysis in **III** and conclusion in **IV**.

## II. Method for Grading Objects using RGB-D Images and CNN

### *A. Transformation for RGB Images Involving Equidistant Objects*

We make RGB images for training and learning as follows; suppose that we have RGB image $G_{\rm ext}^{\rm [RGB]}$ and depth image $G_{\rm ext}^{\rm [Depth]}$ involving an object extracted with the size $W_{\rm ext} \times H_{\rm ext}$ from original RGB-D images, $G_{\rm orig}^{\rm [RGB]}$ and $G_{\rm orig}^{\rm [Depth]}$. Then, we first obtain the 3D point $(x, y, z)$ corresponding to the pixel at $(X_{\rm ext}, Y_{\rm ext})$ with depth value $D_{\rm ext}$ in $G_{\rm ext}^{\rm [Depth]}$ by

$$x = \left(\frac{X_{\rm ext} - 0.5W_{\rm ext}}{W_{\rm ext}}\right)\left(2D_{\rm ext}\tan(0.5\theta_{W_{\rm ext}})\right), \qquad (1)$$

$$y = \left(\frac{Y_{\rm ext} - 0.5H_{\rm ext}}{H_{\rm ext}}\right)\left(2D_{\rm ext}\tan(0.5\theta_{H_{\rm ext}})\right), \qquad (2)$$

$$z = D_{\rm ext}, \qquad (3)$$

where $\theta_{W_{\rm ext}}$ and $\theta_{H_{\rm ext}}$ represent viewing angle of $G_{\rm ext}^{\rm [Depth]}$ (see Fig. 1). In order to recognize an object invariantly with the distance to the object, we replace the 3D points of the object so that the mean position $(x_c, y_c, z_c)$ of the objects for all images is placed at the same position $(0, 0, z_r)$, and then inversely transform to a depth image with the size $W \times H$ by

$$X = \left\lfloor\left(\frac{x - x_c}{2(z - z_c + z_r)\tan(0.5\theta_W)} + 0.5\right)W\right\rfloor \qquad (4)$$

$$Y = \left\lfloor\left(\frac{y - y_c}{2(z - z_c + z_r)\tan(0.5\theta_H)} + 0.5\right)H\right\rfloor \qquad (5)$$

$$D = \lfloor z - z_c + z_r \rfloor \qquad (6)$$

where $\theta_W$ and $\theta_H$ are viewing angles of new image, and $\lfloor\cdot\rfloor$ indicates the floor function. We use $W = H = 100$ and $\theta_W =$

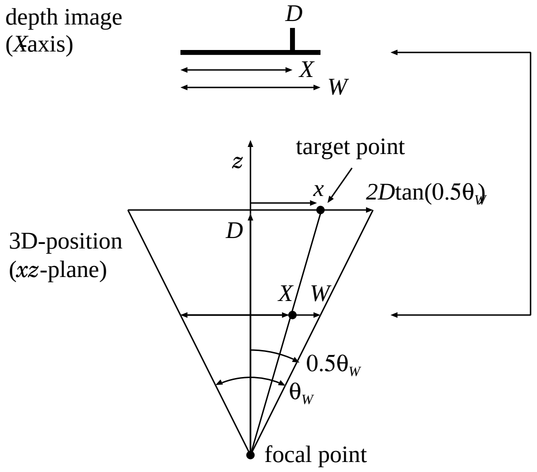


Fig. 1. Relationship between depth $D$ at position $(X, Y)$ in a depth image and 3D position $(x, y, z)$ for $Y = y = 0$. Interpret $D_{\text{ext}} = D$, $X_{\text{ext}} = X$, $Y_{\text{ext}} = Y$, $W_{\text{ext}} = W$ for text.

$\theta_H = 100 \cdot 57/320$ for the data of Kinect v1 in the experiment shown below.

The values $(R, G, B)$ at the position $(X_{\text{ext}}, Y_{\text{ext}})$ in the original extracted RGB image $G^{[\text{RGB}]}_{\text{ext}}$ is used for the transformed image $G^{[\text{RGB}]}$ at $(X, Y)$ as follows;

$$R(X, Y) = R(X_{\text{ext}}, Y_{\text{ext}}) \tag{7}$$

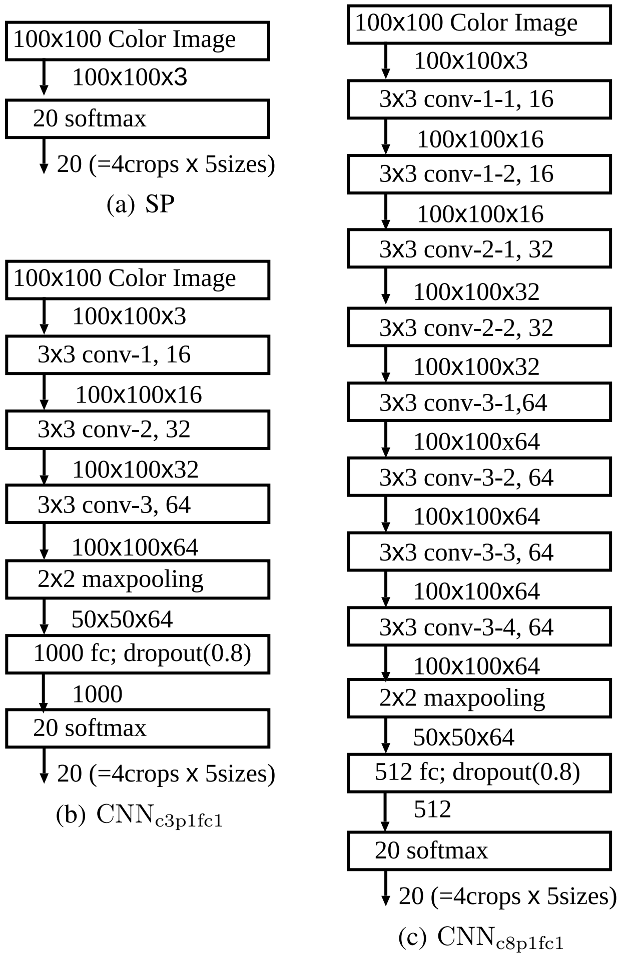
$$G(X, Y) = G(X_{\text{ext}}, Y_{\text{ext}}) \tag{8}$$

$$B(X, Y) = B(X_{\text{ext}}, Y_{\text{ext}}) \tag{9}$$

Here, since the resolution of $G^{[\text{RGB}]}$ and $G^{[\text{RGB}]}_{\text{ext}}$ is different, there is a possibility that a position $(X, Y) \in [0, W) \times [0, H)$ without the correspondence to $(X_{\text{ext}}, Y_{\text{ext}}) \in [0, W_{\text{ext}}) \times [0, H_{\text{ext}})$ exists. Therefore, we have executed linear, quadratic and no interpolation of RGB values for the position without correspondence, respectively, when it is surrounded by two, more than two and less than two positions with correspondence.

### B. CNNs for Grading Objects According to Size

Now, we consider the grading of objects in RGB images according to the size. In research studies on neural network image processing [10], the convolutional neural network consisting of a convolutional layer and a max pooling layer, the cells in the convolutional layer work as learnable filters for extracting local features as simple cells in the visual cortex and the cells in the max pooling layer achieve small shift invariance as complex cells in the visual cortex. Furthermore, size (or scale) invariance is achieved by multiple convolutional and max pooling layers. However, in the grading task, size invariance should not be so large in order for grading the size while allowing small error of transformation arose in the process to obtain RGB images involving equidistant objects shown in the previous section. From this point of view, we have examined three neural networks shown in Fig. 2, where (a) is a simple neural network, which we call SP, and (b) and (c) are CNNs, which we call $\text{CNN}_{\text{c3p1fc1}}$ (involving 3 convolutional layers, a max pooling layer and a fully connected layer) and $\text{CNN}_{\text{c8p1fc1}}$ (involving 8 convolutional layers, a max pooling layer and a fully connected layer), respectively. As learning algorithm, we have used Adam optimizer. Here, note that the structure and the function of the constituent layers, such as convolutional, max pooling, fully connected and softmax layres, and Adam learning optimizer are standard in deep learning architectures [9], [10]. Moreover, we have implemented these neural networks in the Chainer framework [8] as one of the standard frameworks.



(a) SP

(b) $\text{CNN}_{\text{c3p1fc1}}$

(c) $\text{CNN}_{\text{c8p1fc1}}$

Fig. 2. Example of neural networks for comparing the performance in grading objects according to the size. The activation function of the units in all convolutional layers and fully connected layer (fc) are ReLU (rectified linear unit). The dropout ratio for fc layer is 0.8. Output size of each layer is written at the right hand side of the output arc, e.g. “100x100x3” indicates the output size is given by the image consisting of $100 \times 100$ pixcels of 3 channels (RGB).

One of the most important structure is that we have placed only one max pooling layer for the last convolutional layer, or conv-3 for $\text{CNN}_{\text{c3p1fc1}}$ and conv-3-4 for $\text{CNN}_{\text{c8p1fc1}}$. We have obtained this structure as the best one among $8 = 2^3$ combinations of whether we use or not max-pooling layers for convolutional layers (conv-1, conv-2, conv-3 for $\text{CNN}_{\text{c3p1fc1}}$ and conv-1-2, conv-2-2, conv-3-4 for $\text{CNN}_{\text{c8p1fc1}}$). The reason is considered that a max pooling layer reduces the original

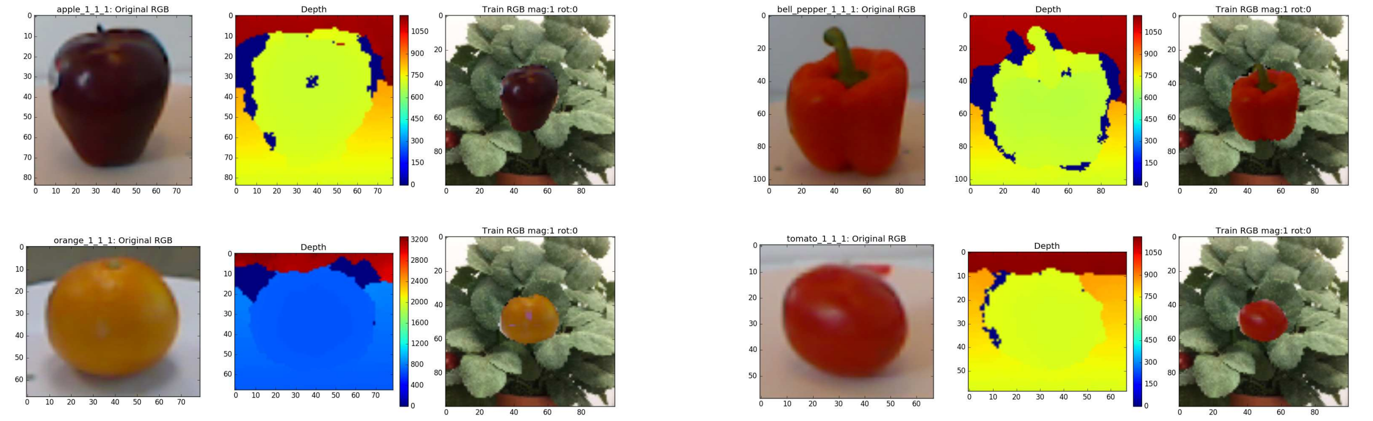

apple_1_1_1: Original RGB
Depth
Train RGB mag:1 rot:0
bell_pepper_1_1_1: Original RGB
Depth
Train RGB mag:1 rot:0
orange_1_1_1: Original RGB
Depth
Train RGB mag:1 rot:0
tomato_1_1_1: Original RGB
Depth
Train RGB mag:1 rot:0

Fig. 3. Example of three images of apple (upper left), bell pepper (upper right), orange (lower left) and tomato (lower right), respectively, where the three images are original RGB image (left), depth image (middle), and training RGB image (right) consisting of transformed object and training background image.

image size from $100 \times 100$ into a half $50 \times 50$ and further size reduction may not be useful for grading objects according to the size.

### C. Learning for Rotation Invariance

In order for rotation invariance, we have trained the networks shown in Fig. 2 with rotated images of the objects. As shown in the experiments described below, we have used the images of objects on a turntable rotated around $y$-axis (or vertical direction), and we rotate the objects in the image around $z$-axis (or depth direction) by transforming the images, while we do not have used the rotation around $x$-axis which is for our future research studies.

### D. Color spaces

For color dependent grading, we have evaluated the performance for three color spaces, RGB, HSV (hue, saturation, and value), and HSL (hue, saturation, and lightness), where HSV and HSL are occasionally used in color detection in robot vision [11], [12].

## III. Numerical Experiments and Analysis

### A. Training and Test Datasets

We have examined the present methods for grading objects with the Washington RGB-D object dataset [13]. This dataset was recorded using a Kinect style 3D camera that records synchronized and aligned 640×480 RGB and depth images at 30 Hz. Each object was placed on a turntable and video sequences were captured for one whole rotation, and has different number of images. Namely, we have used four objects of fruits and vegetables named apple_1_1, bell_pepper_1_1, orange_1_1, and tomato_1_1, each of which has 208, 193, 195, and 249 images, respectively. From the first 180 images for each object, odd-number images are used for training, and even-number images are used for test. From each image, we make five different sizes of objects magnified by the ratio 0.8, 0.9, 1.0, 1.1 and 1.2. Furthermore, each magnified object is rotated randomly in the range $-10$ to 10 degrees around $z$-axis. Thus, we have $5400 = 4 \times 5 \times 3 \times 90$ images in each of training and test datasets. We classify 20 classes consisting of five sizes of four fruits and vegetables (apple, bell pepper, orange and tomato). Thus, each class has 270 images for RGB and depth. For training and test RGB images, different backgrounds of the objects as shown in Fig. 3 and Fig. 4.

### B. Experimental Results and Analysis

We show experimental result of the loss (cross-entropy) and NLACC (negative log accuracy) for training and test dataset vs. the number of learning epochs in Fig. 5. Here, the cross-entropy is defined by

$$E \triangleq -\sum_{n=1}^{N}\sum_{k=1}^{K} d_{nk} \log y_{nk} \tag{10}$$

for the output vector $\boldsymbol{y}_n = (y_{n1}, y_{n2}, \cdots, y_{nK})$ of the network and the target vector $\boldsymbol{d}_n = (d_{n1}, d_{n2}, \cdots, d_{nK})$ w.r.t. the input vector $\boldsymbol{x}_n$ $(n = 1, 2, \cdots, N)$. Here, $d_{nk} = 1$ if $\boldsymbol{x}_n$ is in the $k$th class, and 0 otherwise. On the other hand, the accuracy is defined by

$$\mathrm{ACC} \triangleq \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\left\{ \operatorname*{argmax}_{k=1,\cdots,K} d_{nk} = \operatorname*{argmax}_{k=1,\cdots,K} y_{nk} \right\}. \tag{11}$$

where $\mathbf{1}\{z\}$ denotes an indicator function equals to 1 if $z$ is true, and 0 otherwise. Furthermore, we define $\mathrm{NLACC} \triangleq -\log(\mathrm{ACC})$ for explaining the result from a different point of view of the loss as shown below. When $1-\mathrm{ACC} \ll 1$, we have $\mathrm{NLACC} = -\log(\mathrm{ACC}) \simeq 1/\mathrm{ACC}$, e.g. $\mathrm{NLACC} \simeq 0.1$, 0.01 and 0.001 for $\mathrm{ACC} = 0.9$, 0.99 and 0.999, respectively.

From the training and test loss in Fig. 5, we can see that CNNs ($\mathrm{CNN}_{\mathrm{c3p1fc1}}$ and $\mathrm{CNN}_{\mathrm{c8p1fc1}}$) have achieved better performance than SP from the point of view of the magnitude of fluctuation and ACC achieved at 200 learning epochs. Here, we have reduced the magnitude of fluctuation by a large dropout ratio 0.8 for the CNNs, but we do not have optimized other parameters such as number of units, number of filters, and so on to obtain better performance. Furthermore, we could not have obtained reasonable learning result for RGB and

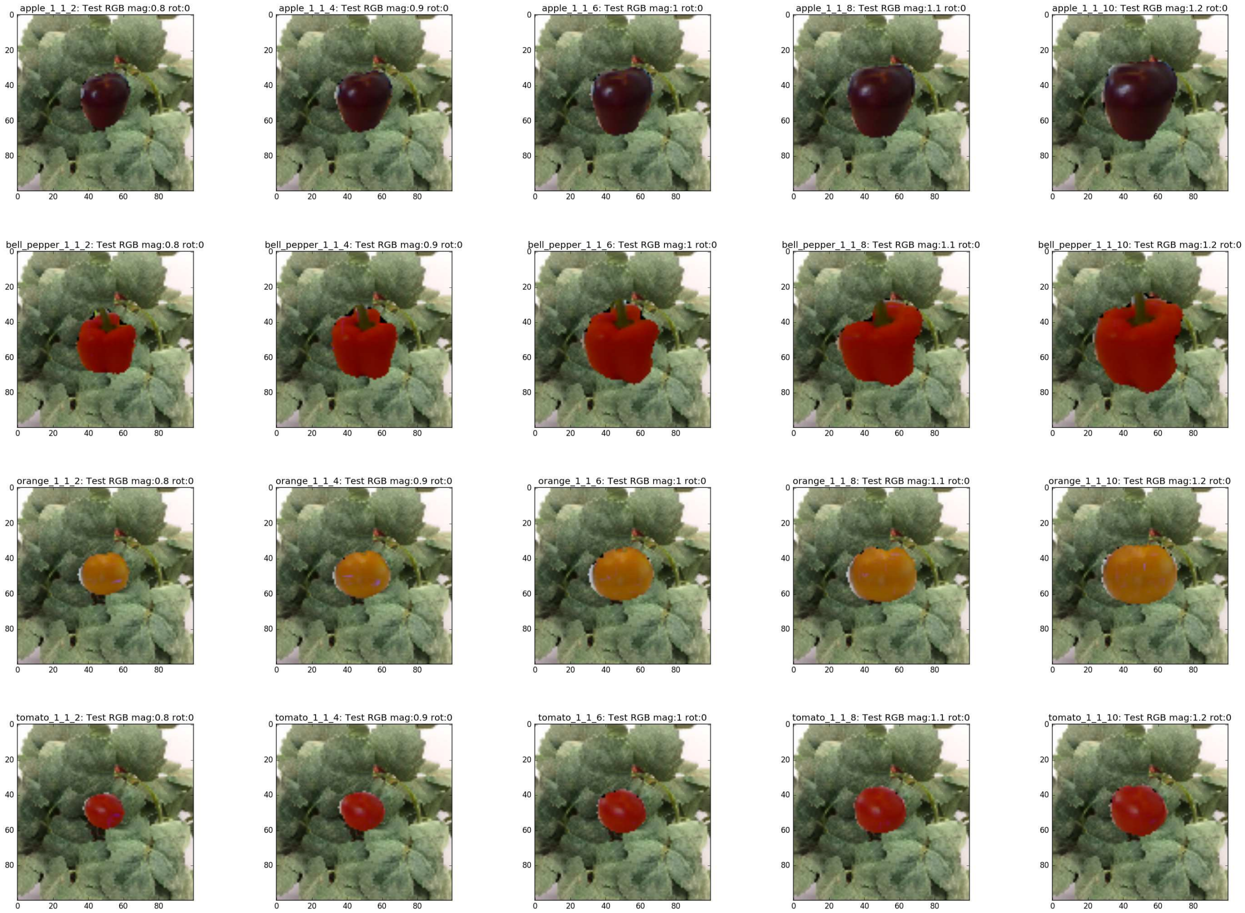

apple_1_1_2: Test RGB mag:0.8 rot:0
apple_1_1_4: Test RGB mag:0.9 rot:0
apple_1_1_6: Test RGB mag:1 rot:0
apple_1_1_8: Test RGB mag:1.1 rot:0
apple_1_1_10: Test RGB mag:1.2 rot:0
bell_pepper_1_1_2: Test RGB mag:0.8 rot:0
bell_pepper_1_1_4: Test RGB mag:0.9 rot:0
bell_pepper_1_1_6: Test RGB mag:1 rot:0
bell_pepper_1_1_8: Test RGB mag:1.1 rot:0
bell_pepper_1_1_10: Test RGB mag:1.2 rot:0
orange_1_1_2: Test RGB mag:0.8 rot:0
orange_1_1_4: Test RGB mag:0.9 rot:0
orange_1_1_6: Test RGB mag:1 rot:0
orange_1_1_8: Test RGB mag:1.1 rot:0
orange_1_1_10: Test RGB mag:1.2 rot:0
tomato_1_1_2: Test RGB mag:0.8 rot:0
tomato_1_1_4: Test RGB mag:0.9 rot:0
tomato_1_1_6: Test RGB mag:1 rot:0
tomato_1_1_8: Test RGB mag:1.1 rot:0
tomato_1_1_10: Test RGB mag:1.2 rot:0
0
20
40
60
80

Fig. 4. Example of test images consisting of transformed objects magnified by the ratio 0.8, 0.9, 1.1 and 1.2 and test background image.

HSV with $\mathrm{CNN}_{\mathrm{c8p1fc1}}$, which is considered to be owing to the vanishing gradient problem and we have applied the batch normalization method but we could not have reasonable result. We would like to solve this problem in our future research studies.

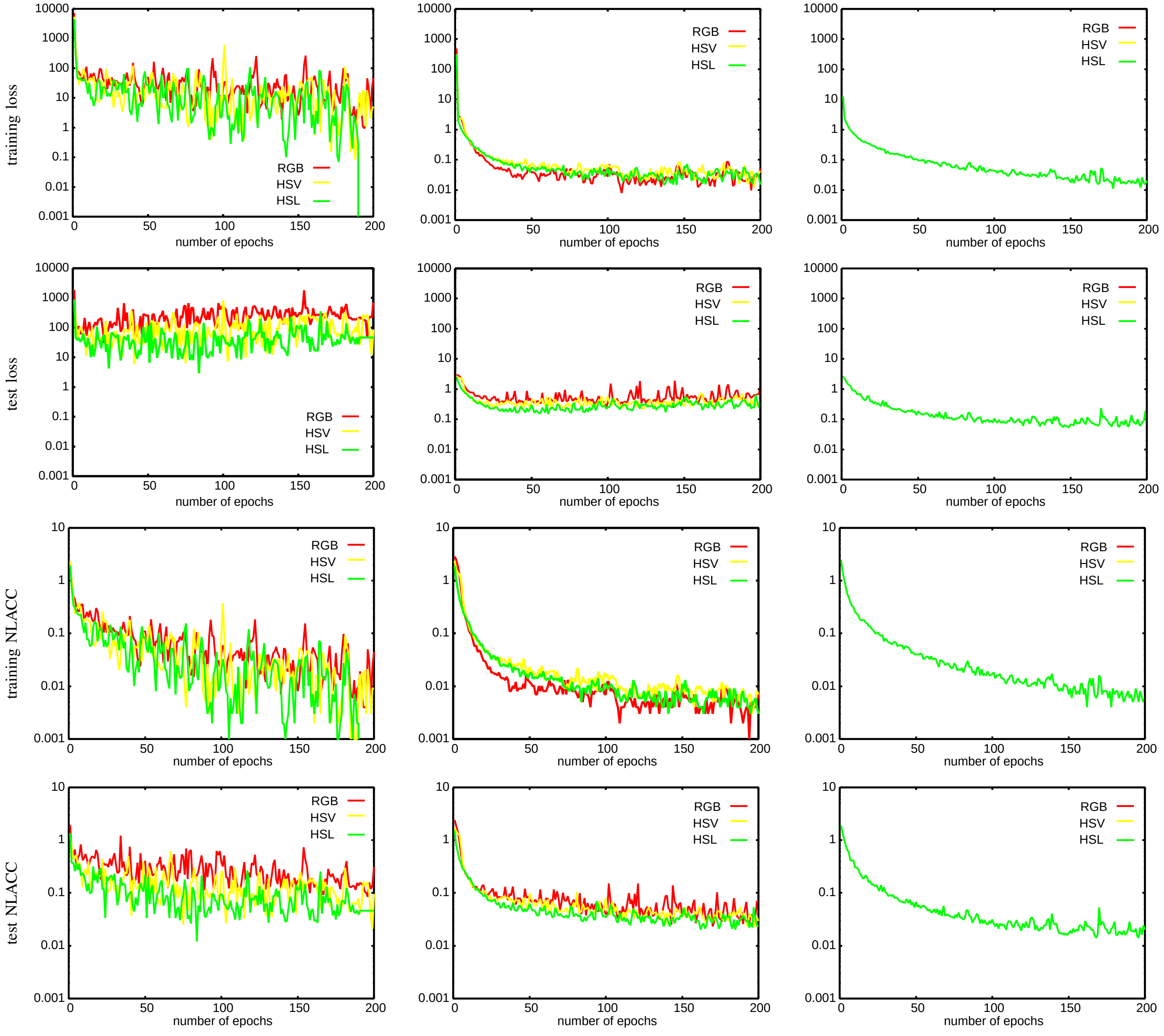
We can observe the overfitting for all cases from the point of view that test loss is larger than training loss. Furthermore, the curve of test loss for $\mathrm{CNN}_{\mathrm{c3p1fc1}}$ and SP increases with the increase of the number of learning epochs, which indicates a phenomenon of increasing overlearning. However, we can see that NLACC decreases with the increase of the number of epochs. Since the learning algorithm is for reducing training loss, this phenomenon is not easy to understand. Since the ultimate objective of learning for this task is to increase ACC or decrease NLACC for test dataset, this phenomenum may indicate a mismatch between learning objective and the learning algorithm for reducing the loss defined by the cross entropy of the output of CNN. From another point of view, we suspect that training and test images are very similar, and the generalization ability of learning algorithm (decreasing test loss) is not so necessary, which indicates inadequate test images. However, the test loss for $\mathrm{CNN}_{\mathrm{c8p1fc1}}$ decreases with the increase of the number of epochs, which indicates that the generalization ability is obtained by modifying the structure of CNN. We would like to clarify this phenomenum in our future research studies.

By means of comparing the performance obtained for RGB, HSV and HSL color spaces shown in Fig. 5, we may say that HSL is superior to HSV and HSV is superior to RGB, while the difference does not seem so large.

In Table I, we show the confusion matrix achieved by $\mathrm{CNN}_{\mathrm{c3p1fc1}}$ at 200 learning epochs. We can see that the performance is good, and the most characteristic grading error occurs for the adjacent sizes for each fruit. Since the CNNs using max pooling layers for conv-1 and conv-2 has provided worse result not in the classification of the name of fruits and vegetables but in the classification of the sizes (not shown in this paper), we can say that max pooling layer with the last convolutional layer plays an important role in grading the objects according to the size.

## IV. Conclusion

We have presented a method of grading fruits and vegetables by means of using RGB-D images and CNN. The method remakes RGB images involving equidistant objects by means of using depth image, and then utilizes CNN for learning to



(a) SP (b) $\mathrm{CNN}_{\mathrm{c3p1fc1}}$ (c) $\mathrm{CNN}_{\mathrm{c8p1fc1}}$

Fig. 5. Experimental result of loss and NLACC for training and test dataset vs. the number of learning epochs. The accuracy, ACC, for test dataset achieved at 200 epochs with HSL, HSV, RGB images is 0.955, 0.909, 0.732 by SP, 0.979, 0.967, 0.929 by $\mathrm{CNN}_{\mathrm{c3p1fc1}}$, and 0.981 by $\mathrm{CNN}_{\mathrm{c8p1fc1}}$ with HSL.

classify RGB images for the grading of objects according to the size, where the CNN is structured for achieving both size sensitivity for grading and shift invariance for reducing error involved in RGB images. By means of numerical experiment, we have shown the effectiveness and the analysis of the method. For easy experiments in this paper, we have generated the images of different size objects by transforming original images, while we are now conducting to use RGB-D images involving actual different sizes of objects as well as actual rotated objects and actual background of objects. Furthermore, we would like to examine the parameters of CNN much more and optimize them for the grading of objects according to not only the size but also other features.

## References

[1] J. Gill, P.S. Sandhu, T. Singh, “A Review of Automatic Fruit Classification using Soft Computing Techniques,” International Conference on Computer, Systems and Electronics Engineering (ICSCEE’2014), pp.91–98, 2014

[2] M. Raj, D. Swaminarayan, “Applications of Image Processing for Grading Agriculture products,” International Journal on Recent and Innovation Trends in Computing and Communication, 3(3), pp.1194–1201, 2015

[3] S. Banot, P.M. Mahajan, “A Fruit Detecting and Grading System Based on Image Processing-Review,” International Journal of Innovative Research in Electrical, Electoronics, Instrumentation and Control Engineering, 4(1), 2016

[4] R. Socher, B. Huval, B. Bhat, C.D. Manning, A.Y. Ng, “Convolutional-Recursive Deep Learning for 3D Object Classification,” Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012

[5] A. Eitel,J.T. Springenberg, L. Spinello, M. Riedmiller,W. Burgard,

TABLE I

Confusion matrix of grading result by $\mathrm{CNN}_{\mathrm{c3p1fc1}}$ with ACC=0.979 obtained at 200 epochs. The name of fruit and vegetable in the first column and the fist row indicate actual and classified name of the object, respectively. The second column and the second row indicate the actual and graded size, respectively.

| | | apple | | | | | bell pepper | | | | | orange | | | | | tomato | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 |
| apple | 0.8 | 270 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.9 | 1 | 268 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.0 | 0 | 4 | 263 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.1 | 0 | 0 | 2 | 266 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.2 | 0 | 0 | 0 | 3 | 265 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| bell pepper | 0.8 | 0 | 0 | 0 | 0 | 0 | 266 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | 0.9 | 0 | 0 | 0 | 0 | 0 | 5 | 256 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 264 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | 256 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 268 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| orange | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 269 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 0.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 270 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 269 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 269 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 269 | 0 | 0 | 0 | 0 | 0 |
| tomato | 0.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 263 | 7 | 0 | 0 | 0 |
| | 0.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 259 | 8 | 0 | 0 |
| | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 4 | 259 | 6 | 0 |
| | 1.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 9 | 249 | 11 |
| | 1.2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 267 |

"Multimodal Deep Learning for Robust RGB-D Object Recognition," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015

[6] C.R. Qi, H. Su,M. Niessner,A. Dai,M. Yan,L.J. Guibas, "Volumetric and Multi-View CNNs for Object Classification on 3D Data," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[7] T.Kamishima , H. Asho, M. Yasuda, S. Maeda, D. Okanohara, T. Okaya, Y. Kubo, D. Bollegala, "Deep Learning (in Japanese)," Kindai Kagakusha, 2015

[8] S. Tokui, K. Oono, S. Hido, J. Clayton, "Chainer: A next-generation open source framework for deep learning," in Workshop on Machine Learning Systems at Neural Information Processing Systems, 2015

[9] T. Okaya, "Machine Learning Professional Series: Deep Learning," Kodansha, Tokyo, 2015 (in Japanese)

[10] T. Kamishima (Ed.), "Deep Learning," Kindaikagakusha, 2015 (in Japanese)

[11] S. Peng, Y. Wen, "Research based on the HSV humanoid robot soccer image processing," Proc. of the International Conference on Communication Systems, pp.52–55, 2010

[12] S.H. Tsai, Y.H. Tseng, "A novel color detection method based on HSL color space for robotic soccer competition," Computers and Mathematics with Applications, 64(5), pp.1291–1300, 2012

[13] K. Lai, L. Bo, X. Ren, D. Fox, "A large-scale hierarchical multiview rgb-d object dataset," Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2011