Industrial Paper

Training Data Generation Method for Deep Learning by Utilizing Computer Graphics

Tsukasa Kudo†, Ren Takimoto†, and Tenma Kawanaka‡

†Faculty of Informatics, Shizuoka Institute of Science and Technology, Japan
‡Shizutetsu Information Center Corpration, Japan
kudo.tsukasa@sist.ac.jp

Abstract - In image recognition, deep learning has enabled innovative improvement in accuracy. As a result, its application has spread to various fields. However, in order to conduct deep learning, it is necessary to accumulate a large number of training data. And, this often becomes the obstacles to applying the deep learning to actual business systems. On the other hand, for computer graphics (CG), various tools have been developed and provided. And, currently, we can not only create a CG image easily but also execute various CG operations automatically by program control. So, in this study, we propose a method to generate the images of training data automatically by using CG. For example, in inventory management of parts, target objects are composed only of inventory shelves and parts with heavy but simple shapes. That is, although it is difficult to accumulate a large number of their actual images, it is expected that these images can be easily generated by using CG. Furthermore, we create CG images for the experiments and automatically generate training data to conduct deep learning for inventory quantities. Then, we conduct experiments to evaluate the estimation accuracy with respect to the inventory quantity shown in the actual inventory photograph. Through this experiment, we show this method is effective in some fields where it is difficult to accumulate a large number of the actual images for the deep learning.

Keywords: Deep learning, Computer grapahics, CG, Inventory management system, Stocktaking, Convolutional neural network

1 INTRODUCTION

Recently, in the field of pattern recognition such as speech recognition and image recognition, the effectiveness of deep learning has been confirmed [17], [13]. As a result, its applications are rapidly spreading to various fields [4], [8]. And, by applying it, recognition accuracy has been rapidly improved; especially in image recognition, the case of achieving accuracy exceeding human vision has also been reported [7], [9], [20]. As one of the reasons for such rapidly spreading, it can be pointed out that a large number of data necessary for learning became to be prepared easily. That is, with the progress of the Internet of Things (IoT), a large number of data such as images and movies are made public on the Internet and data collection also became facilitated [1], [5].

However, there are some fields where the collection of training data is not easy. For example, in the manufacturing factory of mechanical products which our laboratory supports its production management system improvement, the stocktaking of parts inventory is a heavy workload. Especially, since parts quantity in the bulk container cannot be counted from the outside, they must be taken out to count. So, in our previous study, we proposed a method to automatically discriminate the inventory satisfaction by image recognition utilizing deep learning, then constructed and evaluated its prototype. As a result, we found this method is useful, and high reliability can be obtained by comparing with the theoretical inventory obtained by the production management system [12].

For these evaluations, we prepared 1,600 image data for each part. However, in the actual factory, the parts are delivered to the assembly field from parts shelves collectively: by each product lot, namely product manufacturing unit, or by each order composed of several products. So, the number of times of variation of the shelves is comparatively small. For example, in the case where its variation occurs once a day, about 250 images are obtained a year. That is, to accumulate 1,600 images, it takes more than 6 years. In addition, parts are stored in each inventory shelf, and their kinds extend to thousands. And, since most parts are heavy, it is not practical to deliberately change the situation of the inventory shelves by hand so many times. For these reasons, to prepare the training data efficiently has become the problem in applying the deep learning to the image recognition for the actual inventory satisfaction discrimination.

Here, each image of an inventory shelf is composed of only two types of objects: the inventory shelf itself and the part. And, the parts are placed on the shelf according to a certain rule. For example, in the case of storing relatively small parts in a bulk container, the state of the inventory shelf can be composed by piling up the parts randomly from the bottom of the container. Furthermore, the parts have simple shapes such as nuts and bolts, and there are relatively few types of materials for parts. This suggests that a large number of various image data of each inventory shelf can be generated by the computer graphics (CG) efficiently.

The motivation of this study is to show there are fields as follows: it is difficult to accumulate a large number of training data composed of photographs of the actual objects (hereinafter, 'actual images') for the deep learning; but, it is easy to accumulate the training data by utilizing CG. The target of this study is the above-mentioned inventory satisfaction discrimination method in our previous study, which was carried

Figure 1: Inventory shelves of bulk container.

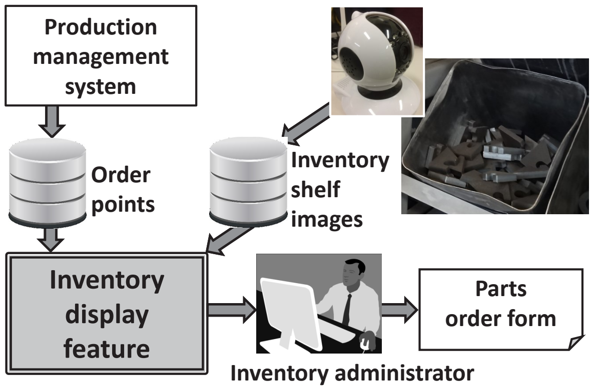


Figure 2: Inventory management utilizing images.

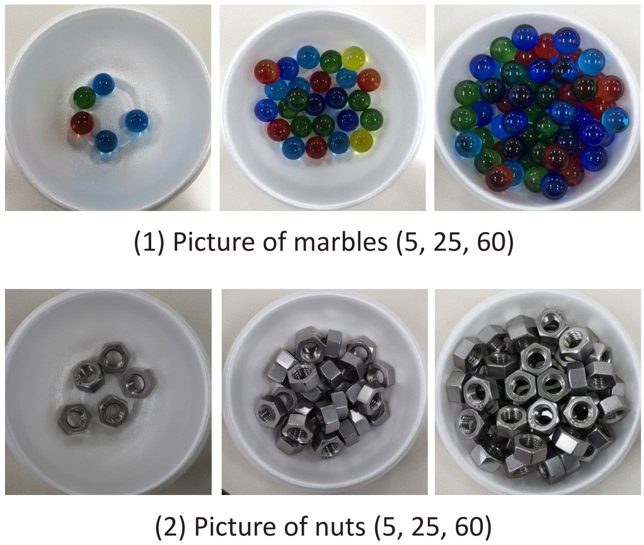


Figure 3: Picture example of experimental objects.

out by using only the actual images for the training data. In this paper, we show the training data generated by CG tool can complement the actual images, that is, the lack of the actual images can be supplemented with the CG images.

Concretely, this paper shows the following four points. The first is the relationship between CG image factors and recognition accuracy, and we show that comprehensive improvement of CG image is necessary to improve the accuracy. The second is the case where both of actual images and CG images are used, and we show the accuracy can be improved by adding some actual images into CG images. The third is the case for different parts, and we show there is a similar tendency with respect to the above-mentioned points. The fourth is the evaluation of man-hours to create CG images of parts on the premise of using the same material and simple shape, and we show it is much more efficient than collecting a large number of actual images.

The remainder of this paper is organized as follows. Section 2 shows the related works and the problem to create the training data, and we propose the training data generation method with CG for the stocktaking in Section 3. Section 4 shows the implementation of this method, and Section 5 shows the experiments and evaluations with the training data generated by this method. We discuss the evaluation results in Section 6 and conclude this paper in Section 7.

2 RELATED WORKS AND PROBLEM

In this section, we explain the background and related works of this study. Our laboratory supports a factory, which manufactures mechanical products, to introduce and operate its production management system. Since thousands of parts in various shapes are stored in each inventory shelf in the factory, the workload required for stocktaking of inventory is a serious problem. In particular, in the case where parts are stored in the bulk container as shown in Fig. 1, they cannot be counted from the outside. So, since it is necessary to take out the parts from the container and count them up, it is a serious factor increasing man-hours.

On the other hand, with the progress of the Internet of Things (IoT), various sensors such as surveillance cameras are controlled remotely, and their data is accumulated and analyzed in the server. As a result, since so various and enormous data has been stored in the database, it has become difficult to deal with such data with conventional relational databases. So, various NoSQL databases have been put to practical use to manipulate such a data efficiently [16]. For example, MongoDB is a kind of document-oriented NoSQL database and provides the GridFS interface to manipulate such data efficiently in the distributed environment [2]. That is, now a day, large capacity of image and video data have become to be easily handled.

Due to such a technical background, we proposed an inventory satisfaction discrimination method using the inventory shelf images shown in Fig. 2 [14]. Here, the ordering point quantity is determined for each part in the inventory shelves by the production management system, and inventory is replenished when its inventory quantity falls below this ordering point [19]. Therefore, to discriminate visually that the inventory of each part satisfies this ordering point quantity is more efficient than performing the stocktaking of the inventory. For example, in the case of the inventory shelf on the upper right of Fig. 1, while it is difficult to grasp its exact quantity, it is relatively easy to discriminate that there are ten or more parts.

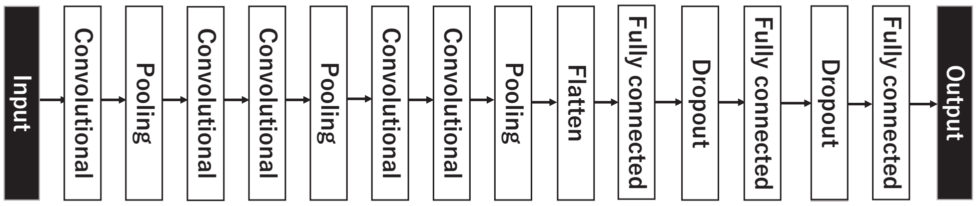


Figure 4: Construction of deep convolutional neural network.

And, in the case where the discrimination was difficult, by replenishing the part from the viewpoint of safety, it became not necessary to count the inventory quantity. As a result, we showed that the efficiency of the inventory management could be achieved by showing both the production plan data and current images of the inventory shelves to the inventory administrator as shown in Fig. 2. However, even by this method, some problems about the workload of the inventory manager remained: he had to check many inventory shelves one by one; especially, in the case where a large number of parts were stored in the bulk container, it took time for the discrimination.

On the other hand, currently, the accuracy of image recognition is rapidly improving by utilizing deep learning. For example, in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), competition in the algorithms for object detection and image classification of large-scale is done. And, after the deep learning was applied in 2012, the recognition rate has been greatly improved. Moreover, it exceeded 5.1% of human recognition rate in 2015 [20]. Along with the improvement in recognition rate, the application of deep learning to image recognition is spreading in various fields such as face authentication, medical image diagnosis, plant disease detection, and so on [4], [8], [15].

So, we conceived to apply the deep learning to the inventory satisfaction discrimination utilizing image recognition. And, as a feasibility study, we conducted the experiment to recognize the classified quantity of the objects by the supervised deep learning as following [12]. We constructed the deep convolution neural network (deep CNN) and performed the deep learning; we evaluated its accuracy of the quantity estimation by using the test data of images prepared separately. For the target objects, we used the marbles and nuts, which is easy to create the training data. And, we classified each of them from 5 to 80 every 5 with the label of their quantity.

Figure 3 shows the image examples of training data in the case of 5, 25 and 60. We prepared 100 image data for each class, that is, the total is 1,600 for each of marble and nut. Then, we padded 400 images by up/down and right/left inversion, and 90 and 270-degree rotation. As a result, we created 500 image data for each class, that is, the total was 8,000 for each of marble and nut. Next, we separated them into 6,000 training data and 2,000 test data, then we separated 600 data from the training data as the verification data. Ultimately, with 5,400 training data, we conducted supervised deep learning with the above-mentioned labels. Figure 4 shows the composition of deep CNN for this deep learning.

After that, we evaluated the accuracy of the quantity estimation in this deep CNN by using the above-mentioned test data. Firstly, we conducted the comparative evaluations between the deep CNN and human vision by using marbles. As a result, we found that although the human vision was able to accurately estimate the quantity in the case where it was small, the deep CNN's accuracy was higher in the case where the quantity was equal or more than 20. This experiment corresponded to the inventory shelf of the bulk container shown in Fig. 1. For example, it was difficult for humans to estimate the quantity of the image of 60 in Fig. 3 in a short time. Furthermore, we evaluated the distribution of the estimated quantity, and we concluded that it could be applied to actual inventory management systems by taking the following countermeasures: the safety stock should be increased to permit the error; the estimated quantity should be compared with the logical inventory quantity calculated by the production management system to detect the error.

In this experiment, since we treated small marbles and nuts shown in Fig. 3, it was easy to obtain a large number of photo to create the training data by shuffling these objects in the bowl at every time with the human hand. However, even with such objects, we took more than one week to take all the photos. Furthermore, in the actual factory, the bulky and heavy parts shown in Fig. 1 are treated. In addition, since it is in operation, we cannot hinder the workers. That is, there is a problem that it is difficult to accumulate a large number of training data for applying the deep learning.

Here, Generative Advisory Network (GAN) has been proposed to generate similar images of actual images, which utilizes deep learning [6], [9]. It consists of the following two parts: the generator network creates images that are intended to come from the same distribution as the training data; and, the discriminator network examines the images to determine whether they are real or fake. And, through training, the former becomes to create fake images that are harder to be discriminated from the real images. In this method, though various similar images can be automatically created, there is a problem that it requires costs of the training and network adjustments.

Generally, to apply the deep learning, accumulating a large number of training data is an important factor to improve its performance, and training data is collected in various ways in each application field. On the other hand, there are application fields where collecting the training data is difficult like this case. Therefore, it is considered effective to develop an

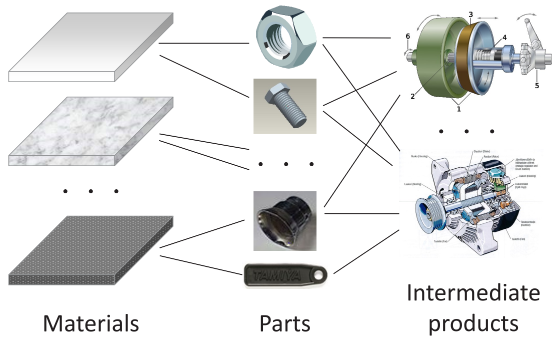


Figure 5: Relationship between materials and parts.

efficient preparing method of the training data in such a field.

3 PROPOSAL OF TRAINING DATA GENERATION METHOD UTILIZING CG

As a solution to the problem in the fields where the accumulation of a large number of training data is difficult, we propose a training data generation method by utilizing CG. Considering the characteristics of the inventory shelf, usually, only one kind of parts are stored in one shelf. That is, from the viewpoint of CG, it is possible to construct the state of the inventory shelf with only two objects, one inventory shelf and one part. Furthermore, it is not necessary to consider the deformation of each object. That is, by piling up the part objects randomly in the inventory shelf object, the state of the inventory shelf can be constructed virtually.

In addition, as shown in Fig. 5, in the case of machine parts, although the types of parts are several thousand, the number of types of material is relatively few such as iron, stainless steel and so on. For example, it is less than 20 in the target factory. Also, each part has a simple shape as shown in Fig. 1, and these are combined to produce the 'intermediate products' in Fig. 5; and, the final product is assembled by using them. Therefore, from the viewpoint of CG modeling, namely creating CG objects of parts, only a simple shape processing is the main work. Incidentally, we use the same objects as our previous study, namely marbles and nuts shown in Fig. 3. And, we perform the comparative evaluations of the accuracy between the cases of utilizing the CG image and actual image in deep learning.

Basically, to construct such inventory shelves by CG, the four factors should be considered: the shape of the objects, the material of the objects, the illumination as the environment, and the placement of each object on each inventory shelf. Firstly, the shapes of objects can be realistically created by using CG modeling tools, based on the measuring data of the actual inventory shelf and part. Secondly, the material expresses the textures of these objects, and it can be also added by the CG modeling tools. However, it is necessary to adjust the material by not only human vision, but also checking the influence to the deep learning. Thirdly, the illumination needs to be determined based on the actual factory illumination environment, and it can be added in the same way as the material. These three factors are static with respect to each inventory shelf. So, it can be used repeatedly after once created.

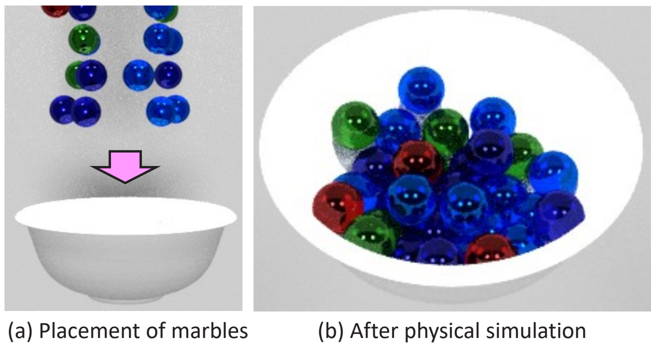


Figure 6: Image example of experimental objects.

Fourthly, the placement of the part objects is the most important factor to create a large number of training data, and the following three requirements should be considered. The first is a physical requirement, for example, it is necessary that the placement in CG does not collapse even if it is actually placed. The second is the realization of random placement. That is, in order to accumulate a large number of training data for images of the same part, different arrangements must be made for each image, even for the same number of parts. The third is the rule of placement. For example, the parts in the second container from the left of the uppermost shelf of Fig. 1 are the one like plates, and they piled for every 4.

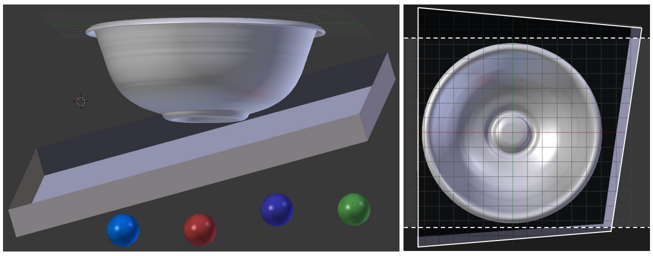
Regarding such a placement, various kinds of CG tools are provided. And, many of them can be controlled automatically by the programming language and provide the feature of physical simulation such as free-fall by gravity. Therefore, it is possible to create the various placement state similar to the real world by executing the physics simulation after randomly placing parts by using programming languages.

For example, we show the case to generate the marble training data shown in Fig. 3 by CG tool. Firstly, as shown in (a) of Fig. 6, we create the bowl and marble objects, then place the marble objects randomly above the bowl. Next, the marble objects are dropped in the direction of the arrow shown in (a) of Fig. 6 by using the physical simulation of free-fall by gravity. After falling in the bowl, they converge to the natural state shown in (b) by the control of the physical simulation. And, by saving the rendering image of this result, the image shown in Fig. 3 can be obtained. By repeating these processes by using the programming language, various large number of training data can be created automatically.

As described above, it is expected that a large number of training data can be generated efficiently by using CG tools in the following case: the target images are composed of a small number of objects; and, a large number of different image data can be created according to a certain procedure. As a result, it is considered that the training data can be generated with CG images as the substitute for actual images in a certain field, where it is difficult to accumulate the training data by actual images.

```
> blender --background --python Marble_basic.py
```

Figure 7: Batch file to control Blender by Python.



(a) Initial placement of objects in Blender (b) Camera view

Figure 8: Initial placement of objects in Blender.

4 IMPLEMENTATION OF TRAINING DATA GENERATOR

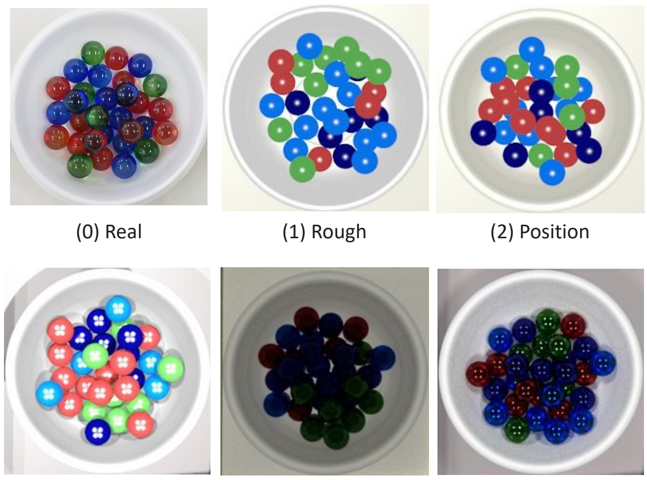
In this study, we used 3DCD creation software Blender 2.79 [3] to generate the training data. Blender can be controlled by the Python script, and Python 3.5.3 is shipped with the above-mentioned Blender. So, after we place the necessary objects and lights in Blender, the arbitrary number of training data can be generated automatically by using Python program according to the processes shown in Section 3. We created Blender objects and Python programs separately and executed them with the batch file shown in Fig. 7.

(a) of Fig. 8 shows the placement of objects in Blender, which is composed of a bowl, four marbles, and a square tray. In this experiment, same as the previous study, we used four types of marbles. So, we placed each one under the tray. And, the tray is used to detect the spillage of marble objects from the bowl object. That is since there is a possibility that the marble objects spill out in the physical simulation in the case where the number of them is large, such an image must be excluded. (b) of Fig. 8 shows the view from the camera for the rendering. The white solid quadrangle is the outline of the tray, and the lower right is the lowest position. So, the spilled marble objects gather here. Then, the spillage can be detected by comparing the image before and after the simulation after trimming the bowl object. Incidentally, as for the illumination, we placed Hemi lamp, which lights up the whole area equally, and a directional lamp on the side of the camera; the range between white dashed lines in (b) of Fig. 8 was the rendering area.

The procedure to generate the images for the training data by using the batch file in Fig. 7 was as follows. Firstly, the Blender file was opened and the marble objects in (a) of Fig. 8 were randomly selected, then their copies were placed hierarchically as shown in (a) of Fig. 6. Here, to shorten the falling time, each hierarchy was divided into four quadrants, and marble objects were placed randomly within the range of each quadrant. Next, the state of (b) in Fig. 6 was generated by physical simulation. Then, rendering was performed by the camera placed right above the bowl object as shown in (b) of Fig. 8, and its image was saved in a file. After that, the Blender file was reopened to return to the initial state, then the same procedure was repeated.

Table 1: Experiment cases on marbles.

No	Case	Position	Illumination	Material
(1)	Rough			
(2)	Position	○		
(3)	Illumination	○	○	
(4)	Material	○		○
(5)	All	○	○	○



(0) Real (1) Rough (2) Position

(3) Illumination (4) Material (5) All

Figure 9: Images of marbles by photography and CG.

After all the image data were generated, they are converted to the training data by another Python program by the following procedures. Firstly, images with spilled marble objects were excluded, then the outside of the bowl object in the remained images was trimmed to create images similar to Fig. 3. Then, the designated number of images were saved as the training data. After that, by the same procedure as the previous study as shown in Section 2, the images were padded 4 times to create the final training data.

5 EXPERIMENTS AND EVALUATIONS

In the experiments, firstly we evaluate each factor of creating CG images from the viewpoint of the influence on the accuracy of image recognition, by using marbles. Secondly, we evaluate the change of this accuracy according to the ratio of replacing a part of the CG images with the actual images. Thirdly, we evaluate the above-mentioned tendency for the object with different shape and material, by using nuts. In the above experiments, we use the CG images created in the procedure shown in section 4. Lastly, we evaluate the man-hours to create a part CG image only targetting the shape processing.

5.1 Evaluations of CG Images for Marbles

Table 1 shows the factors of creating the CG image, which was evaluated in the experiments: '(2) position', '(3) illu-

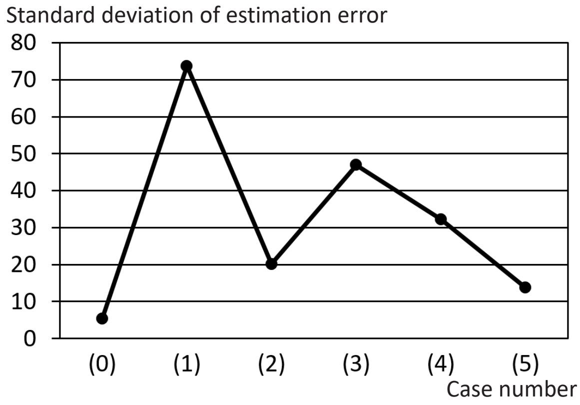


Figure 10: Change in standard deviation of errors.

mination' and '(4) material'. We evaluated the influence on the accuracy of each factor, and the case of combining them which is shown in '(5) all'. Incidentally, in Table 1, '(1) rough' is the state before improvement of each factor. Figure 9 shows the examples of the images of the training data of 30 marbles: '(0) real' shows the actual image and the others show the CG images created with each case of Table 1. Since marble objects were randomly placed in Blender in each case, the placements of the marbles were not the same.

In '(1) rough' of Fig. 9, since bowl modeling accuracy was low, the placement of marbles expanded and there were more gaps between marbles than the one in '(0) real'. So, in '(2) position', we improved the shape of the bowl, and the placement of marbles became closer to (0). Next, since the actual images were taken in a room where multiple fluorescent lamps were installed on the ceiling, we placed multiple lamps in Blender to make the CG images close to the actual environment in '(3) illumination'. In '(4) material', we improved only the material of the marbles from (2). In Blender, since it was necessary to change the rendering engine from 'Blender rendering' to 'Cycles rendering' in order to produce the marble material, the brightness of the whole image changed in (4). In '(5) all', we added the lamps of (3) to the CG image of (4).

Next, we trained the deep CNN shown in Fig. 4 with these training data. And, we prepared the test data composed of 50 actual images for each case of 5, 20, 40, 60 and 80 marbles. Then, we obtained the estimated quantity of each test data by the deep CNN. Figure 10 shows the standard deviation of the estimation error of each case of Table 1. This error is the difference between the actual quantity in each image and the estimated quantity using the deep CNN. Here, 'case number' corresponds to 'No.' in the Table 1. Although the standard deviation of '(0) real' was 5.3, the one of '(1) rough' worsened to 73.7. And, it was bettered to 20.1 by improving the position as shown in (2). On the other hand, each improvement of '(3) illumination' and '(4) material' worsened than (2). Here, each improvement was applied separately. However, in the case of applying both shown in '(5) all', it bettered to 13.8. That is, it was necessary to improve the illumination and material together.

Also, Fig. 11 shows the distribution of errors of estimated quantities with respect to each actual number of marbles in each case of between '(0) real' and '(5) all'. Here, since we

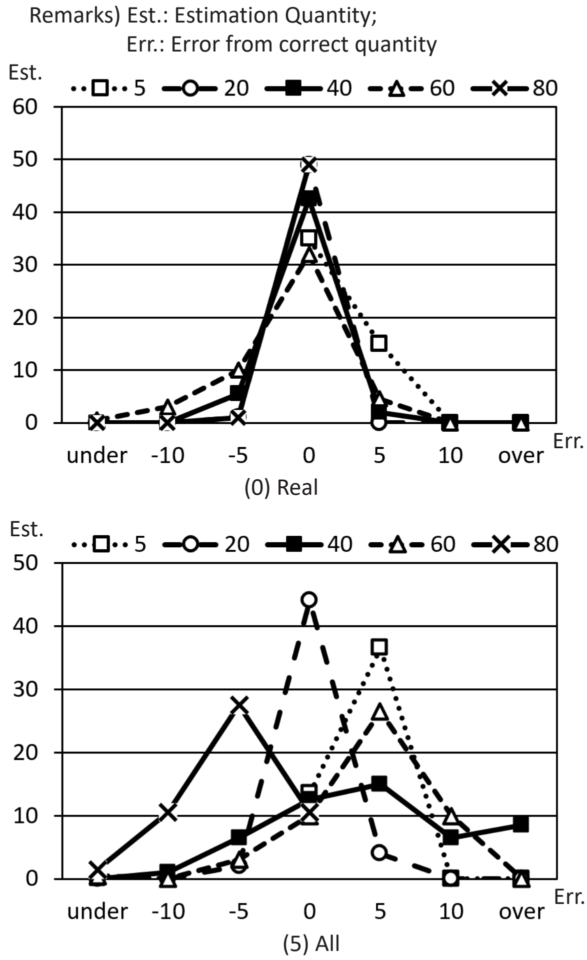


Figure 11: Distribution of errors of estimated quantities.

classified the number of marbles from 5 to 80 every 5 and trained by the supervised learning, the error also changed in units of 5. In (0), the error is distributed in the range of ± 10 around the correct '0'; while in (5), many peaks of distribution appeared before and after '0'. However, even in the latter case, though the error 'over' namely 15 or more occurred for about 20% images in the case of 40 marbles, the error was within the range of roughly ± 10 in the other cases.

5.2 Evaluations of CG and Actual Mixed Images

The creation of high-precision CG image similar to the actual images requires a large number of man-hours. Conversely, the actual images can be obtained even in the case of the above-mentioned inventory shelves in the factory, if the number of images is small. So, we evaluated the change of the accuracy in the case where the CG images are replaced with the actual images.

We used CG images of '(5) all' in Table 1 in this experiment, and the designated number of images were replaced with the actual images. For example, in the case of '5%', we prepared 500 images using 425 CG images and 75 actual images. In addition, the test data for evaluation are the actual images same as Section 5.1.

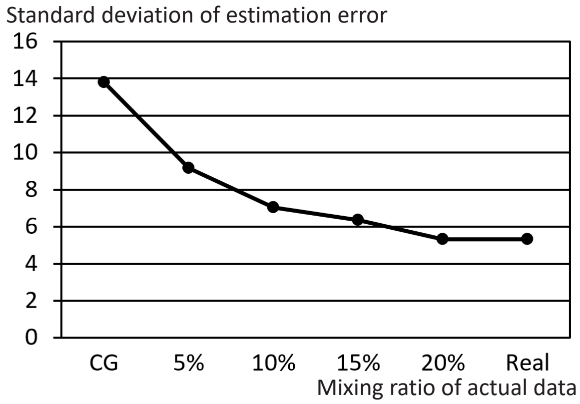


Figure 12: Standard deviation of errors on mixing ratio.

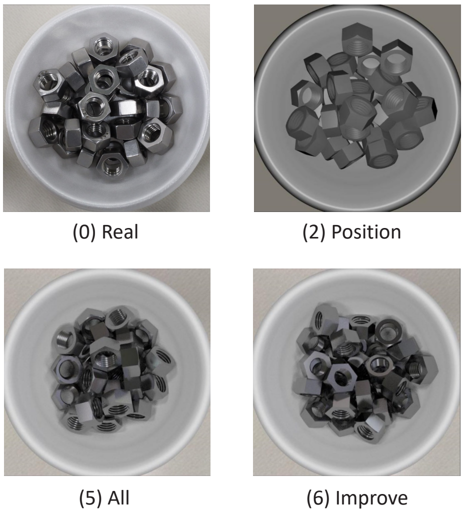


Figure 13: Images of nuts by photography and CG.

Figure 12 shows the change of the standard deviation of errors according to the mixed ratio of the actual data from 5% to 20% for every 5%. The standard deviation improved to about 2/3 when 5% of the CG images were replaced with the actual images, which was about 55% improvement as for the difference between (5) and (1). Similarly, as of 10%, it improved to about half, which was about 80% for the difference between (5) and (1). Furthermore, as of 20%, the standard deviation became almost the same as of the actual images.

That is, in the case of replacing the CG images with the actual images, the improvement of the accuracy was relatively larger than the replacement ratio.

5.3 Evaluations in Nuts

In order to confirm that even for the objects with different materials and shapes, they have the trends similar to those shown in Sections 5.1 and 5.2, we conducted the same experiment as that shown in Figs. 10 and 12. As shown in Fig. 3, the nut has a difference in material as well as shape from marbles, that is, it is made of metal.

Figure 13 shows examples of CG image of 30 nuts created for this experiment. Incidentally, '(0) real' is an example of actual image same as that in Fig. 9. Firstly, based on the results in Sections 5.1, we created '(2) position' that does not take material into consideration and has rough shape. This corresponds to (2) in Table 1, though the illumination is set similarly to '(5) all' in Table 1. Then, we created '(5) all' with a comprehensive improvement of material, lighting, and shape, similar to '(5) all' in Table 1. Lastly, since the shape of the nut is more complicated than the marble, we improved the shape again and created '(6) improve'.

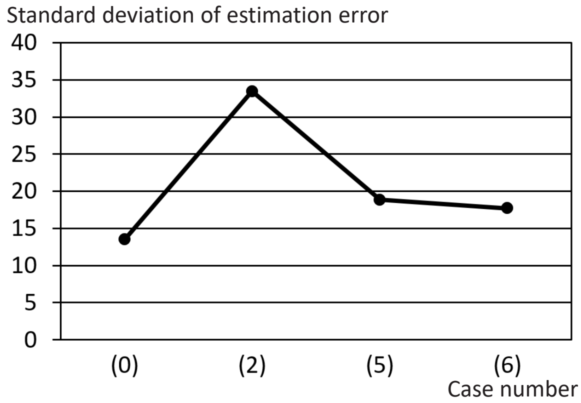


Figure 14: Change in standard deviation of errors (nuts).

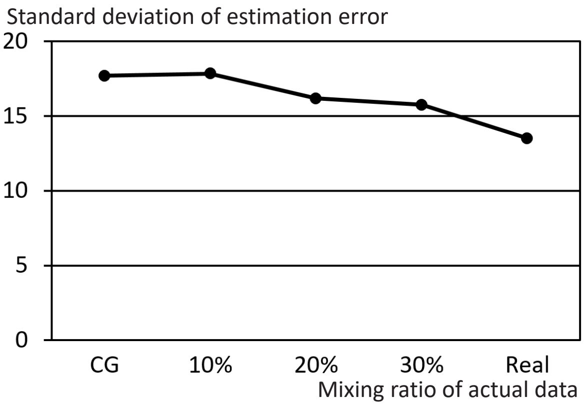


Figure 15: Standard deviation of errors on mixing ratio (nuts).

We trained the deep CNN with these data. Then, similar to Section 5.1, we prepared the test data with actual images of nuts and obtained the estimated quantity for each of them. Figure 14 shows the standard deviation of the estimation error, and we obtained the same tendency as that in Fig. 10. That is, it was 13.5 in the case of (0); it worsened to 33.5 in (2), and improved to 18.8 and 17.7 in (5) and (6) respectively.

Next, we evaluated the change of the accuracy in the case where the CG images are replaced with the actual images, similar to Fig. 12. As shown in Fig. 15, similar to the case of marbles in Fig. 12, accuracy was improved by replacing the CG images with actual images. However, its improvement rate was smaller than that of marbles.

5.4 Evaluations of Creation Efficiency of Part CG

Since factories handle a large number of parts, it is necessary to prepare these training data with CG images. There-

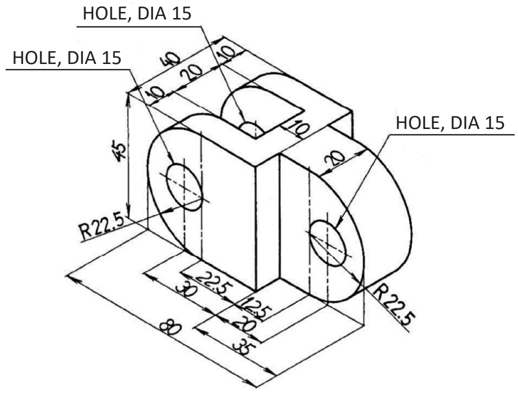


Figure 16: Drawing of machining. [11]

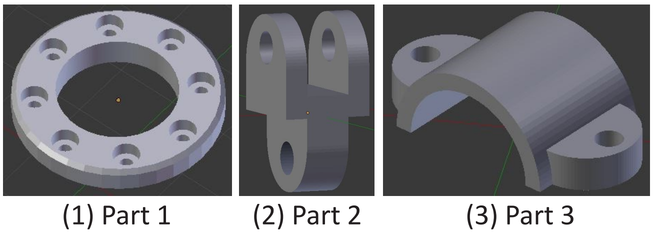


Figure 17: Parts modeling with CG tool.

fore, their quantity to be created also becomes large. On the other hand, from the viewpoint of CG image, there are few kinds of materials as shown in Fig. 5, and surrounding environment such as illumination is roughly the same. That is, as for one material, if there is a CG model of one part, it is only necessary to model the shape for the other part. Then, its training data can be created automatically by the procedure shown in Section 3. And, for the parts, there are drawing of machining shown in Fig. 16.

Therefore, on the premise of the same material and illumination as in Section 5.2, we evaluated the working time of parts modeling with the CG tool based on the drawing shown in Fig. 16. Figure 17 shows the CG model created in this experiment. Incidentally, '(2) part 2' is based on Fig. 16. Here, since target parts shapes are generally simple as shown in Figs. 1 and 5, we used drawings of a rudimentary machining [11].

Figure 18 shows the working time to create these models. Here, although the worker had some experience using

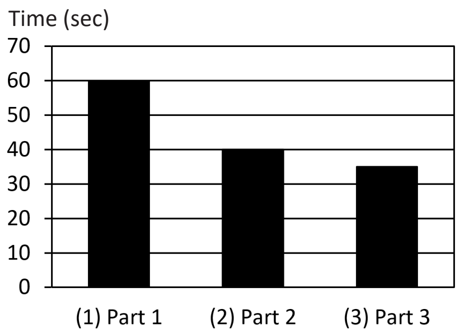


Figure 18: Working time for part modeling.

the CG tool Blender, he had no experience of parts modeling other than the above-mentioned nut, nor mechanical drawing. Therefore, the working time does not include the time used for interpretation of the drawing or consideration of the creation method. As shown in '(1) part 1' of Figs. 17 and 18, although the time increased as the processing place increased, the working time was 60 minutes at most.

That is, the working time to create the CG model was so smaller than that to take the picture mentioned in Section 2.

6 DISCUSSION

In this study, we evaluated the accuracy in the case of using the CG images instead of the actual images for the training data of the deep learning. Firstly, we examined the case of the marbles. As a result, as shown in Fig. 11, we found that a certain degree of accuracy could be achieved, while the errors spread larger compared to the case of the actual images. On the other hand, as shown in the relation between the CG images of Fig. 9 and the standard deviations of the error in Fig. 10, the accuracy was greatly affected by the factors to create the CG images.

In other words, it is expected that the accuracy can be improved by refining the CG image shown in (5) of Fig. 9 to make it closer to the actual images shown in (0). However, the more we refine the CG images to look like the actual images, the more the workload becomes big. That is, it is necessary to repeat the several works accompanied by trial and error: the first is the adjustments of the CG images such as material and illumination; the second is the deep learning of the deep CNN. And since each takes a long time, the ratio of improvement to cost is expected to decline.

Therefore, we showed that the accuracy could be improved efficiently by using the training data, in which a part of the CG images was replaced with the actual images, as shown in Fig. 12. That is, we found that in the case where the training data was generated with the CG images are used, it is effective to prepare the actual images in possible and to use them as a part of the training data. In addition, at the factory like the target of this study, the operation to estimate the inventory quantities by using the deep learning as follows is possible: initially, the training data is composed of the CG images and a few actual images; then, while increasing the actual images in operation, the training of the deep CNN is repeated by using these actual images and the accuracy can be gradually improved.

Here, to improve efficiently the accuracy obtained by using CG images, it is considered to use plural methods in combination: the proposed method in this paper, and another method, for example, a method using GAN shown in Section 2. By using the proposed method, the CG models of parts could be created accurately and efficiently from the drawings of machining as shown in Figs. 17 and 18; and, the placement of parts could be automatically determined by the physical simulation shown in Fig. 6. Therefore, since training like GAN is not necessary, we consider that this method is more efficient than GAN as for these processes. On the other hand, for example, to adjust the materials and illuminations of parts shown in Figs. 9 and 10, we had to perform trial and error.

So, we consider that the accuracy may be improved more efficiently by applying such as GAN to these processes, and this is one of the themes of the next study.

Furthermore, we examined the case of the nuts and obtained a similar result to the case of marbles. So, we consider that it is possible to prepare the training data for the deep learning by using CG images for parts of various materials and shapes. However, as shown in Figs. 12 and 15, we found that there was the following difference between the case of marbles and nuts. One was the value of the standard deviation of the error; Other one was the improvement ratio in the case of replacing the CG images with the actual images. And with regard to the cause of the latter, we considered that it was due to the ratio of errors between the case of using CG images and actual images, which was smaller in nuts compared to marbles.

Lastly, we evaluated the working time of CG image creation, we confirmed that the training data could be created far more efficiently than collecting actual images in the case of target parts. Therefore, it is considered that even the fields where it is difficult to accumulate a large number of actual images for the deep learning, there are fields where the target is composed of the simple objects from the viewpoint of CG. In such a case, we consider that to use the CG images generated automatically for the training data is effective.

However, in this study, we have verified the effectiveness of the proposed method by using only the marbles and nuts at just the laboratory. So, the verification in the actual factory remains as the future study. That is, it is necessary to evaluate the accuracy in the case of utilizing the CG images of the various shapes of parts in the actual inventory shelf environments such as shown in Fig. 1.

7 CONCLUSION

Currently, the accuracy of image recognition utilizing the deep learning is rapidly improving, and such an image recognition is applied to various fields. On the other hand, in order to improve the accuracy by the deep learning, it is necessary to accumulate a large number of training data. And, the preparation of the training data often becomes the obstacle to applying the deep learning.

For this problem, we proposed a training data generation method by utilizing CG in this study. And, we created the CG model and confirmed that a large number of training data could be generated by using the CG tool automatically; a certain degree of the accuracy could be achieved by using such a training data. Also, we found the accuracy could be improved by replacing a part of these training data with the actual images. Furthermore, we confirmed training data could be created much more efficiently by this method than collecting the actual images in the case of simple objects such as mechanical parts. As a result, we conclude it is effective to generate the training data by using the CG images in some fields, where it is difficult to accumulate a large number of the actual images for the deep learning.

Future study will focus on the confirmation of its effectiveness in actual inventory environment, namely by using the actual parts in the actual factory, and the method to improve the accuracy efficiently, especially due to the material and illumination.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer networks, Vol. 54, No. 15, pp.2787–2805 (2010).

[2] K. Banker, "MongoDB in Action," Manning Pubns Co. (2011).

[3] Blender Foundation, "Blender," https://www.blender.org/ (reffered, May 2018).

[4] T. H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?," IEEE Transactions on Image Processing, Vol. 24, No. 12, pp. 5017–5032 (2015).

[5] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," Mobile networks and applications, Vol. 19, No. 2, pp.171–209 (2014).

[6] F. Chollet, "Deep learning with python," Manning Publications Co. (2017).

[7] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," 2012 IEEE conference on computer vision and pattern recognition (CVPR), pp. 3642–3649 (2012).

[8] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," IEEE Transactions on Medical Imaging, Vol. 35, No. 5, pp. 1153–1159 (2016).

[9] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, "Deep learning," MIT press (2016).

[10] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," arXiv preprint arXiv:1701.00160 (2016).

[11] M. Higuchi, "Drafting basics," http://www3.nit.ac.jp/ mhiguchi/seizu-01.pdf (referred Oct. 10, 2018) (in Japanese).

[12] T. Kawanaka, and T. Kudo, "Inventory Satisfaction Discrimination Method Utilizing Images and Deep Learning," Procedia Computer Science, Vol. 126, 937-946 (2018).

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton,"Imagenet classification with deep convolutional neural networks," Proc. advances in neural information processing systems, pp. 1097–1105 (2012).

[14] T. Kudo, Y. Ito, and Y. Serizawa, "An Application of MongoDB to Enterprise System Manipulating Enormous Data," Int. J. Informatics Society, Vol. 9, No. 3, pp. 97–108 (2017).

[15] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in plant science, Vol. 7, Art. 1419 (2016).

[16] E. Redmond, and J.R. Wilson, "Seven Databases in Seven Weeks: A guide to Modern Databases and the NoSQL Movement," Pragmatic Bookshelf (2012).

[17] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," Proc. twelfth annual conference of the interna-

tional speech communication association, pp. 437–440 (2011).
[18] E. A. Silver, D. F. Pyke, and R. Peterson, "Inventory management and production planning and scheduling," John Wiley & Sons (1998).
[19] S. P. Singh, "Production and Operation Management," Vikas Publishing House Pvt Ltd (2014).
[20] M. Verhelst, and B. Moons, "Embedded Deep Neural Network Processing: Algorithmic and Processor Techniques Bring Deep Learning to IoT and Edge Devices," IEEE Solid-State Circuits Magazine, Vol. 9, No. 4, pp. 55–65 (2017).

(Received October 10, 2018)
(Revised February 11, 2019)

Tsukasa Kudo received the BSc and ME from Hokkaido University in 1978 and 1980, and the Dr. Eng. from Shizuoka University in 2008. In 1980, he joined Mitsubishi Electric Corp. He was a researcher of parallel computer architecture and engineer of business information systems. Since 2010, he is a professor of Shizuoka Institute of Science and Technology. Now, his research interests include deep learning and database application. He is a member of IEIEC and IPSJ.

Ren Takimoto is currently working toward a B.I. degree at Shizuoka Institute of Science and Technology, Japan. His research interests include computer graphics design.

Tenma Kawanaka received a B.I. from Shizuoka Institute of Science and Technology, Japan, and joined Shizutetsu Information Center Corporation in 2018. His research interests include deep learning. His contribution to this paper was carried out in his graduation research.