Regular Paper

CG Training Model Application Method Using Cycle-consistent Adversarial Network

Tsukasa Kudo†

†Faculty of Informatics, Shizuoka Institute of Science and Technology, Japan
kudo.tsukasa@sist.ac.jp

Abstract - Deep learning-based image recognitions have achieved high accuracy; however, its application to many actual business systems remains difficult. First, deep learning requires extensive training data, which are often difficult to obtain for particular businesses. Second, actual business environmental scenes change owing to various factors, such as the time of day, season, and weather. Consequently, typically, there is a domain gap between the training and target images, which often reduces recognition accuracy. To address these problems, a method to automatically generate training data that is not affected by scene changes using computer graphics (CG) has been proposed. However, domain gap problem between the CG generated images and actual images remains. Recently, Cycle-Consistent Adversarial Networks (Cycle-GAN) have been proposed to translate an image from one domain to a fake image of another domain. In this study, a method to use a model trained with CG images is proposed. In this method, actual images are translated into fake CG images using a Cycle-GAN. An experimental evaluation demonstrates improved accuracy; the proposed method is applied to an inventory estimation of parts in a bulk container using deep learning regression model.

Keywords: deep learning, GAN, Cycle-GAN, regression model, stock-taking, computer graphics, CG, image recognition

1 INTRODUCTION

Currently, deep learning has been applied effectively to various fields [6], [4]. Its effectiveness has been demonstrated in the ImageNet Large Scale Visual Recognition Challenge [12]. And, image recognition accuracy has improved rapidly [19]. With improved accuracy, deep learning has been used with a wider range of image processing applications, such as face recognition, medical image analysis, and plant disease detection [20], [8], [17]. Additionally, an increasing number of images have been stored on cloud servers; therefore, obtaining sufficient images for training data has become easier. However, obtaining sufficient images as training data for individual businesses remains difficult and time-consuming.

In previous studies, I have investigated stock-taking in a machine factory. Typically, in a machine factory, most parts are stored in bulk containers, and for stock-taking, counting the parts manually through a superficial visual examination is impossible. Thus, to count the parts, they must be removed from the container, which needs heavy work-load. To address this problem, I investigated applying deep learning to inventory estimation using image recognition and confirmed that practical accuracy could be achieved using a deep learning regression model [14]. This study was performed in a laboratory where images of lightweight marbles and nuts were used for the training and test data; 1,600 original images were increased to 8,000 by padding.

However, applying this method to an actual factory was difficult. First, to capture images for training data, heavy parts must be repositioned manually. Additionally, typically, there are thousands of bulk containers in a factory, and capturing images of the parts in each bulk container is not practical. Second, since the scene in an actual factory changes due to various factors, such as the time of day, season, and weather, there may be a domain gap between the training and target images. Such domain gaps can reduce recognition accuracy [21].

To address the first problem, we have previously developed a method to generate sufficient training data automatically using computer graphics (CG) [13]. In this study, I propose an inventory estimation system that uses CG generated training data and target data created from images of actual parts. To eliminate the domain gap between the actual and CG images, the actual images are translated to fake CG images before the inventory estimation. Thus, it is expected that inventory estimation accuracy can be maintained regardless of the change of scenes.

Here, this translation is performed using a Cycle-Consistent Adversarial Network (Cycle-GAN) [24], which is a type of generative adversarial network (GAN) [6]. A Cycle-GAN model is trained using different image groups, such as zebras and horses, where images of one group (zebras) are automatically translated into fake images of another group (horses). For some types of objects, it has been shown that images can be translated into highly accurate fake images using this trained model.

In this study, the objective is to verify the feasibility of using a Cycle-GAN to improve the estimation accuracy of the above-mentioned inventory estimation system [13]. It comprises a regression model trained with CG images, and the target data of actual images.

The remainder of this paper is organized as follows. Section 2 reviews related works, the author's previous studies, and states the primary goal of this study. In Section 3, a method to estimate inventory using a training data generator and Cycle-GAN is described. The effectiveness of using fake CG images translated by a Cycle-GAN is evaluated in

Figure 1: Parts inventory in bulk containers

Section 4. Evaluation results are discussed in Section 5, and conclusions and suggestions for future work are presented in Section 6.

2 RELATED WORK

In this section, the author's previous studies and related studies are described. Also, the objective of the current study is discussed. The parts inventory in a machine factory is managed based on a theoretical inventory planned using a production management system. However, actual and theoretical inventories often differ due to product defects, work errors, work delays and so on. Therefore, manual stock-taking must be undertaken. However, manual stock-taking needs heavy workload because, typically, parts are stored in bulk containers (Fig. 1) and to obtain an accurate count of individual parts, the parts must be removed from containers.

On the other hand, in recent years, with the progress of deep learning, the accuracy of image recognition has improved and image recognition has been applied to various fields [8], [12], [17], [19], [20]. Therefore, my colleague and I developed and evaluated a method to estimate inventories using images taken from the outside of the bulk containers. First, we evaluated results obtained using a deep learning multi-class classification model and found that it could achieve a certain estimation accuracy.

Also, we demonstrated how this method could be applied to actual inventory management [10]. The method produced some estimation errors; however, we found that the number of out-of-stock parts could be reduced by increasing the safety stock according to the error range. Also, accuracy could be improved by comparing the estimated inventory with the theoretical inventory. Note that in the multi-class classification model, inventory quantities are treated as discrete classes, and an inventory estimated from a bulk container image is considered a discrete class (quantities) [4].

However, preparing training data for that study was extremely time-consuming. Lightweight marbles and nuts were used to represent parts; 1,600 different original images were captured by repeatedly rearranging these items manually. The 1,600 images were padded to create 8,000 images; 6,000 were used as training data, and 2,000 were used as test data. However, in an actual factory, using this process to obtain training and test data is not practical because typical parts are extremely heavy and there are thousands of bulk containers.

Therefore, I developed a training data generator and determined that a model trained using CG generated data could achieve a certain accuracy [13]. It became evident that based on a machine drawing of the part, the shape of the CG model of the part could also be created relatively easily. In contrast, given various textures and illumination environments, a trial and error process was required to obtain the color tone of parts. Note that, compared to the types of parts, the number of textures was small because the parts were made using limited types of materials, such as iron and aluminum.

In the multi-class classification model, training data must be prepared for each inventory quantity. That is, when inventory quantities fluctuate over a wide range, it is necessary to prepare a large amount of training data. Therefore, accuracy was evaluated using a regression model [14], which estimates an inventory as a continuous quantity [4]. As a result, compared to the multi-class classification model, high estimation accuracy was obtained when actual images were used for both the training and test data. Also, similar accuracy could be obtained for plural parts when estimation was performed with CG images using models trained with CG images.

However, when estimation was performed with actual images using the regression model trained with CG images, estimation accuracy was reduced significantly due to differences in color tone between the actual and CG images, i.e., the domain gap. Other studies have also found that image recognition accuracy deteriorates due to the domain gap between training and target images [21]. This suggests that applying deep learning to stock-taking in an actual factory will be problematic because factory conditions are not stable. Consequently, the domain gap occurs in both the training and target images.

Besides, in 2015, the automatic generation of images using DeepDream, which is an image processing method that uses deep learning, was reported. Since then, studies on automatic image generation and image style conversion by applying deep learning have been actively conducted [4], [18]. It has been shown that a GAN method can be used to generate a fake image from a genuine image. This method comprises a generator network (hereinafter, 'generator') that generates a fake image and a discriminator network (hereinafter, 'discriminator') that classifies images as fake or genuine [7]. By training both networks together, it is possible to generate a fake image that is close to the genuine image. Various methods to generate fake images based on the GAN have been proposed [15], such as pix2pix, DiscoGAN and UNIT [9], [11], [16].

Among such methods, a Cycle-GAN can be applied to translate an image from one domain into an image of another domain. For example, an image of zebras is translated into a fake image of horses by training a Cycle-GAN model with the image group of zebras and horses [24]. Figure 2 shows the structure of a Cycle-GAN model, where X and Y show different groups of images such as zebras and horses.

In Fig. 2, image x of group X is translated to a fake image $\hat{y}$ of group Y by generator G, and discriminator D_Y determines whether $\hat{y}$ is a genuine or fake image of Y. Further-

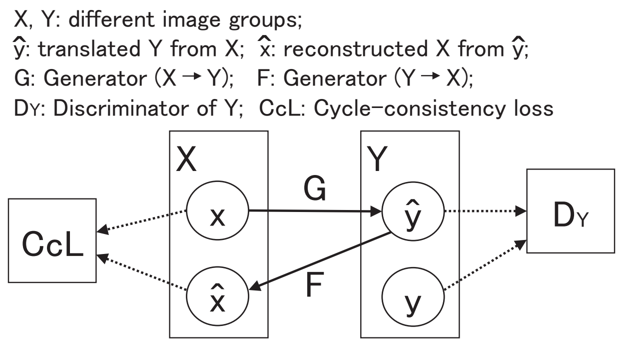

Figure 2: Structure of Cycle-GAN model

more, $\hat{x}$ is reconstructed from $\hat{y}$ by generator F, and Cycle-consistency loss, which is the loss between x and $\hat{x}$, is evaluated. In this study, the feasibility of applying Cycle-GAN to solve the domain gap problem between training and target data in regression models is evaluated.

I assume a case where the Cycle-GAN model is used for quantity estimation from images, such as stock-taking of parts in bulk containers. In other words, I attempt to show that estimation accuracy can be improved by translating actual images to fake CG images using Cycle-GAN. Here, Fig. 2 shows the case of translating X to a fake image of Y. Similarly, images of Y are also translated to fake images of X, and the discriminator and cycle-consistency loss evaluation are performed. In this way, similar to a GAN, by training the generator and discriminator together, images of one group can be translated into high-precision fake images of another group.

Also, given the structure of the Cycle-GAN model (Fig. 2), image translation can be performed automatically between unrelated image groups X and Y. Therefore, by applying the Cycle-GAN model to the above-mentioned inventory estimation, actual images of bulk containers taken under various scene conditions can be automatically translated into fake CG images generated by the training data generator. Most importantly, since CG images can be generated as in a fixed environment, it is expected that the domain gap problem can be solved.

Currently, Cycle-GAN has been applied to data translation in various fields such as videos, voices, and human images [2], [5], [21]. Also, various image translation models have been proposed using the concept of Cycle-GAN, such as Combogan, Sem-GAN, CYC-DGH, and CinCGAN [1], [3], [22], [23]. However, to the best of my knowledge, no study has applied Cycle-GAN to image recognition and evaluated accuracy improvement by eliminating the domain gap between the training and target images.

In this study, the feasibility of applying Cycle-GAN to solve the domain gap problem between training and target data in regression models is evaluated. I assume a case where a regression model trained with CG images is used for quantity estimation from actual images, such as stock-taking of parts in bulk containers. In other words, I attempt to show that estimation accuracy can be improved by translating actual images to fake CG images using Cycle-GAN.

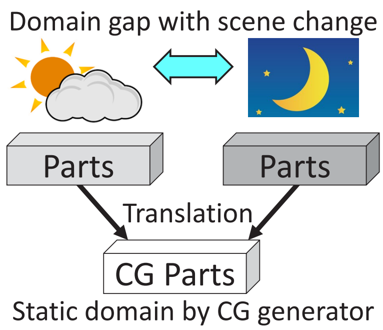

Figure 3: Domain gap in factory and generation of a static domain images

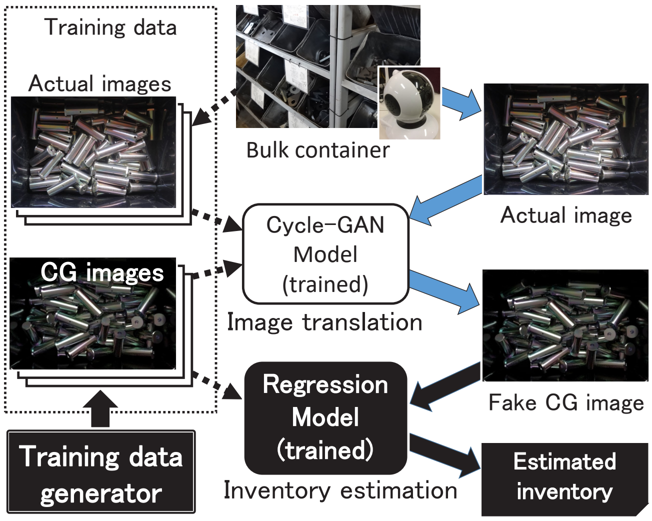

Figure 4: Inventory estimation system dataflow

3 INVENTORY ESTIMATION PROCESS USING TRAINING DATA GENERATOR

Section 2 discussed the two problems when applying deep learning to parts inventory estimation in an actual factory, i.e., preparation of a large amount of training data and domain gap among images due to scene changes. The method shown in Fig. 3 is proposed to address these issues. First, a large amount of training data ('CG parts' in Fig. 3) is generated by the training data generator using CG mentioned in Section 2 under specified conditions. Second, the target parts images ('Parts' in Fig. 3) in the factory are translated into fake images of CG parts (fake CG images). As a result, the domain gap between the training and target images is eliminated.

Figure 4 shows the data flow of the inventory estimation system, in which a Cycle-GAN is used to translate actual images into fake CG images. As shown in the dashed box in Fig. 4, training data for the Cycle-GAN model are prepared as follows. Actual images are created from pictures of factory bulk containers, and CG images are generated using the training data generator. This system uses the regression model trained with CG images, as shown in the black hatched round box shown in Fig. 4. This regression model comprises convolutional layers with pooling layers and fully connected layers, as shown in Fig. 5. Note that the mean square error (MSE) is

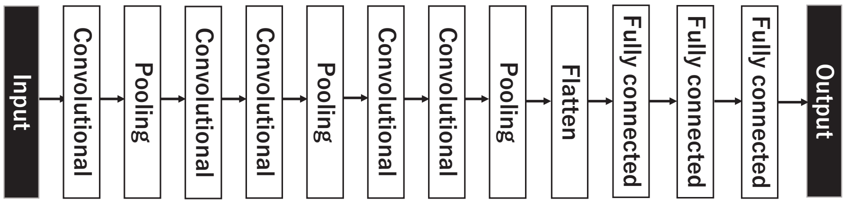

Figure 5: Structure of regression model

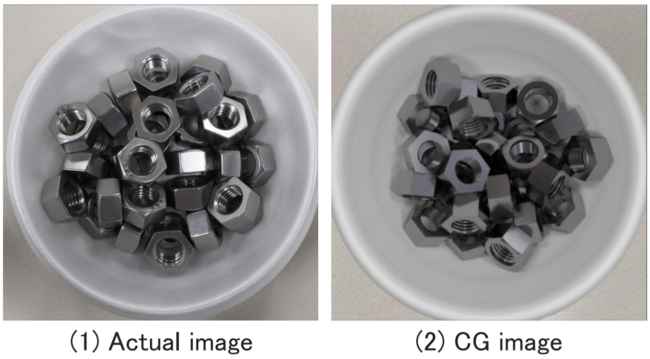

Figure 6: Target nut images

used for its loss function.

As shown in Fig. 4, the actual image of each bulk container is captured by a camera installed around the containers, and the inventory is estimated from this image. To suppress the increase in the inventory estimation error due to the domain gap between the actual and CG image, the captured image is translated into a fake CG image using the Cycle-GAN model. Then, the regression model estimates inventory quantity using this fake CG image.

For the regression model's training data, the shape of the part is important relative to maintaining estimation accuracy. Also, a single camera monitors many bulk containers; thus it is necessary to create training data images not only from just the above direction but also from the direction of the camera for each bulk container. Using the training data generator, the shape of the CG part model can be created easily based on a drawing of the machining of the target part, and the camera position of the CG can be designated at rendering based on the actual camera position. In other words, the training data generator can generate a large amount of training data automatically for the parts in each bulk container.

Therefore, the Cycle-GAN model is only used to eliminate the domain gap due to the color tone of the actual and CG images. It primarily depends on the scene change shown in Fig. 3 and part texture due to its material. Here, there are relatively few types of part materials as mentioned in Section 2, thus Cycle-GAN model training is performed for only each of these materials. In other words, for the training data, CG images are generated by the training data generator, and the actual images are created by collecting images of parts made of the same material. As discussed in Section 2, relative to the Cycle-GAN training data, it is not necessary to associate an individual actual image with an individual CG image. Thus, it is possible to accumulate actual images without investigating part quantities.

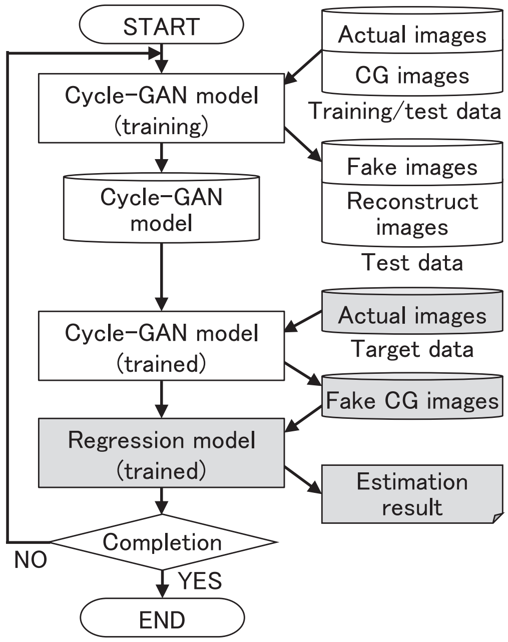

Figure 7: Structure of experimental system

4 EXPERIMENTS AND EVALUATIONS

4.1 Experimental Environment

Figure 4 shows the experimental environment used to evaluate the effect of translating actual images to fake CG images using the Cycle-GAN in inventory estimation. First, images of nuts placed in a bowl were used as the target parts, which were captured from above as shown in Fig. 6. For example, Fig. 6 (1) shows an actual image created from the picture of the nuts, and Fig. 6 (2) shows a CG image generated by the training data generator. Here, 100 images were prepared for each nut quantity from five to 80 for every five, for each of these (1) and (2). In total, 1,600 images were prepared and divided into training (1,200 images) and test (400 images) data.

The structure of the experimental system is shown in Fig. 7. In this experiment, the Cycle-GAN model was trained using training data comprised of actual and CG images. The training was performed as batch processing, and the number of batch per epoch was 1000. Here, discriminator loss (d_loss) and cycle-consistency loss (g_loss) were monitored every 200 batches. Simultaneously, the test data (i.e., actual and CG images) were translated into fake images. Then, each original image was reconstructed from each fake image. Figure 8

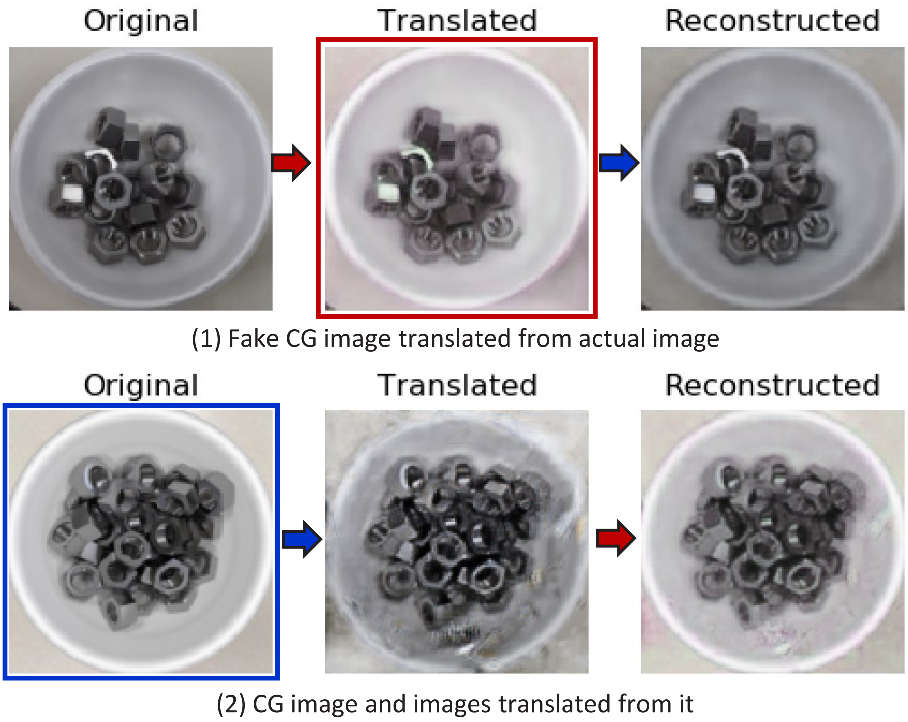

Figure 8: Images generated by Cycle-GAN model

shows examples of these data.

The Cycle-GAN model was saved, and the target data for inventory estimation, which were actual images prepared separately, were translated into fake CG images using this saved model. Here, the actual images comprised 16 types of quantity groups, similar to the training and test data for the Cycle-GAN model, i.e. five to 80 for every five. Also, 50 images were prepared for each group, i.e., 800 images in total. To evaluate the estimation error automatically, the number of nuts in each image was added as a correct label.

The experimental system was implemented on a PC with Windows 10 using Python and Keras. Note that TensorFlow was used for the Keras backend, and OpenCV was used for image conversion. The images were converted to 128×128 pixels and used for training and estimation. For the training data generator, a nut and bowl were modeled using the Blender (a 3DCG modeling tool). Here, the nuts were placed into the bowl using Blender's physical simulation function. Note, Blender's physical simulation and rendering processes were automated using Python.

To training the Cycle-GAN model, its hyper-parameters were set as follows. λ was set to 10.0, which is the strength of the cycle-consistency loss against discriminator D_Y (Fig. 2) and D_X (discriminator for generator F). The argument of the Adam optimizer was set to $Adam(0.0003, 0.5)$. Similarly, for the regression model, the reduction rate of the learning rate was 0.1, the minimum learning rate was set to 10^{-10}, dropout was not used, the output dimension of the fully connected layers was 128 (except for the last layer), and the best model in the training transition was saved. Note that the output dimension of the last layer of a regression model is one.

4.2 Evaluations

Figure 9 shows the transition of d_loss and g_loss of the Cycle-GAN model and the MSE of the inventory estimated by the regression model according to the training progress of the Cycle-GAN model. As can be seen, the MSE decreased as training progressed and became the smallest at batch number 1000 of epoch 12. Then the MSE increased. Note that d_loss fluctuated greatly before the MSE became the smallest and became a very small value several times at batch number 1000 of epoch 8, batch number 600 of epoch 5 and so on. However, there was no tendency for the MSE to improve at these times. Similarly, no clear correlation between g_loss and MSE was observed.

Figure 8 shows an example of the input and output images of the Cycle-GAN model when the MSE became the smallest value. Figure 8 (1) shows (from the left) the actual image, the fake CG image translated from the actual image, and the reconstructed image obtained using the fake CG image. Each corresponds to x, $\hat{y}$, and $\hat{x}$ in Fig. 2. Similarly, Fig. 8 (2) shows a CG image, its fake actual image, and the reconstructed CG image. Inventory estimation was performed by the regression model using the fake CG image at the center in Fig. 8 (1). The original image at the left in Fig. 8 (2) is a CG image and the translated fake CG image in (1) was generated as a fake image of this CG image.

Next, to evaluate the effect of images translated using the Cycle-GAN on inventory estimation, i.e., the effect of fake CG images, we performed a comparative evaluation using CG images, fake CG images, and actual images. Figure 10 shows the MSE results for these image types. As can be seen, using fake CG images, MSE improved by approximately 2.8 times

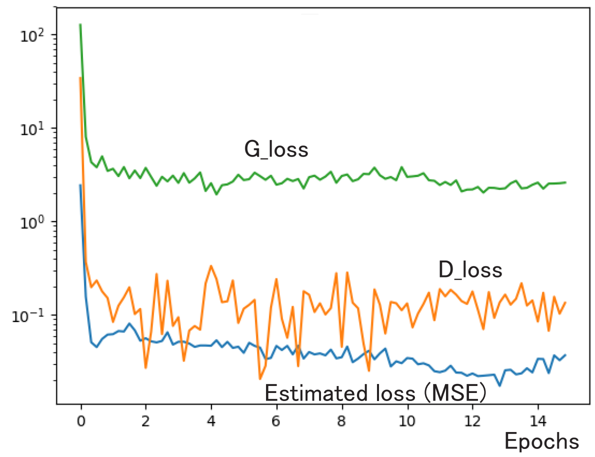

Figure 9: Transition of MSE in inventory estimation with Cycle-GAN training

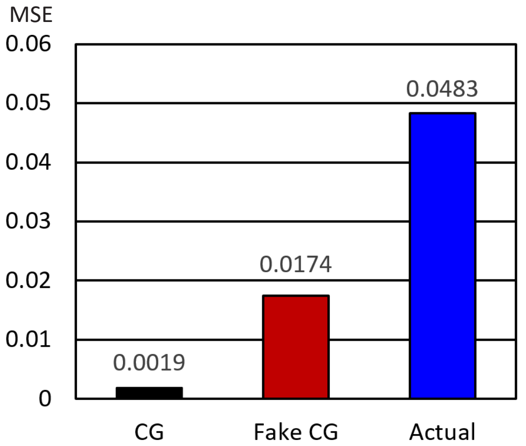

Figure 10: MSE of estimated inventory with different image types

compared to using actual images. However, MSE deteriorated by approximately 9.2 times compared to using CG images.

Figures 11, 12 and 13 show histograms of the estimation errors for CG images, fake CG images, and actual images. In each figure, the horizontal axis shows the error; thus the position of 0 is the correct estimation quantity. Note that 'under' gives the total number of images for which estimation error was -15 or less, and 'over' represents the case for 15 or greater. Besides, the quantities were estimated by the regression model; thus estimation errors were also decimal values. Therefore, the graph in Figs. 11, 12 and 13 was created after rounding off estimation errors to integers. The vertical axis shows the rate of the number of occurrences of each error, and each figure shows cases of five, 20, 40, 60, and 80 nuts as shown in the legend.

As shown in Fig. 11, in the case where inventory was estimated using CG images using the regression model trained with CG images, the distribution of estimation error was approximately within ± 5. Conversely, as shown in Fig. 13, when using actual images for the same model, the distribution was spread across a wider range, i.e., between -10 and over. Note that errors became over in the 40 and 60 nuts cases. As

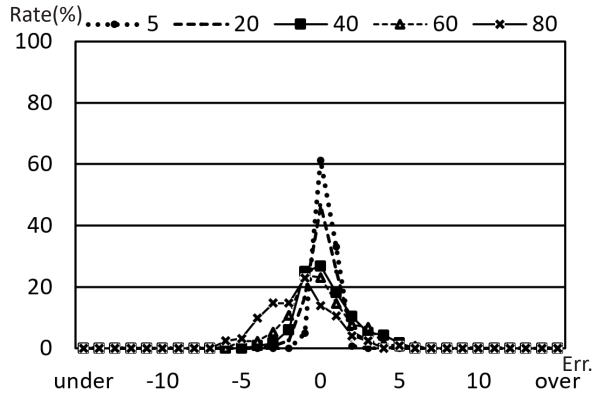

Figure 11: Histogram of error distribution with CG images

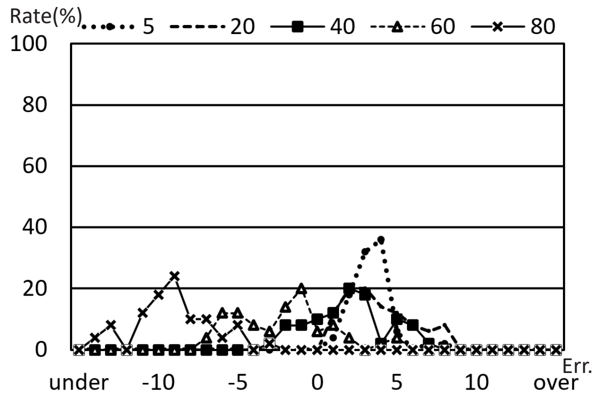

Figure 12: Histogram of error distribution with fake CG images

shown in Fig. 12, when using the fake CG images, the error distribution improved (except the case of 80 nuts) compared to using the actual images, i.e., the distribution was within ± 9. However, deviation to the negative direction increased with 80 nuts.

Furthermore, for inventory estimation using fake CG images, the magnitude and deviation of the errors were evaluated for each quantity of nuts using the mean absolute error (MAE) and the average error. Figure 14 shows the transition of MAE and average error with increasing nut quantity. Here, the vertical axis shows the errors, and the horizontal axis shows nut quantity. In the range of five to 20 nuts, both the MAE and the average error were approximately 4. At 25 nuts, deviation in the positive direction increased. When nut quantity was over 25, the deviation in the negative direction increased linearly as nut quantity increased.

5 DISCUSSION

In a previous study, the estimation accuracy of parts inventories in bulk containers using the regression model was evaluated using the following data: CG data generated by the data generator were used for training the model, and actual images were used to estimate inventory using the trained model. As a result, although training data generation efficiency improved, the problem caused by the domain gap of images occurred and estimation accuracy deteriorated. Furthermore, it

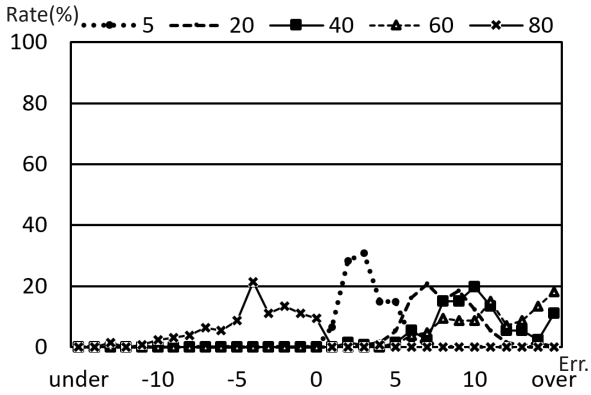

Figure 13: Histogram of error distribution with actual images

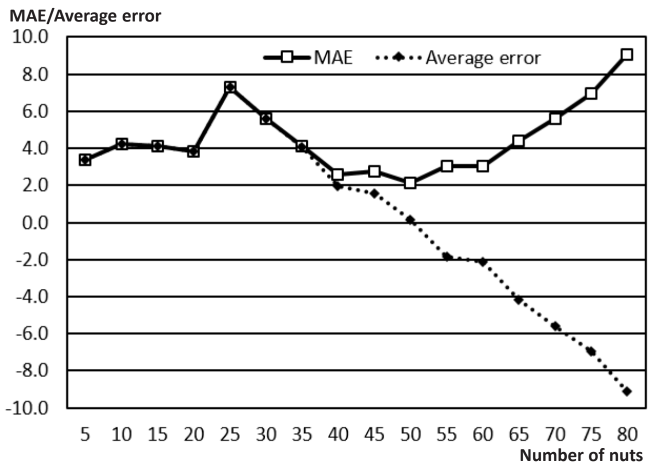

Figure 14: Transition of MAE and average error

was predicted that the same problem would occur due to scene changes in an actual factory. Therefore, to address this problem, the feasibility of translating actual images to fake CG images using a Cycle-GAN was investigated in the current study.

First, I performed comparative evaluations of estimation accuracy with and without image translation using the Cycle-GAN model. As shown in Fig. 10, this image translation process was effective relative to improving estimation accuracy. In particular, as shown in Fig. 14, when the quantity of nut was 20 or fewer, the MAE of the estimation was approximately 4. In actual factories, one of the most important purposes of stock-taking is to prevent running out-of-stock. For this purpose, bulk containers with small inventory quantities are targeted. As discussed in Section 2, deep learning techniques can be applied to actual factory environments by increasing the safety stock and collating with the theoretical inventory. For example, as shown in Fig. 12, when the inventory quantity was 20 or fewer, the error was 10 or less. Therefore, out of stock can be suppressed by increasing the safety stock by 10. Incidentally, it has been confirmed that the types of materials of parts were limited; thus, it is considered that the estimation method using a Cycle-GAN can be applied effectively to multiple parts. Therefore, it is expected that this method can be applied practically in some field even with its current accuracy.

And, two issues related to applying the Cycle-GAN model have been identified. The first issue is related to the method by which the optimal Cycle-GAN model is detected. As shown in Fig. 9, in the range of epochs I have experimented with, strong correlations between the estimation accuracy (MSE) of the regression model and the loss (d_loss, g_loss) of the Cycle-GAN model could not be observed. In other words, from the viewpoint of generating optimal fake CG images for the regression model, the optimal model could not be detected by monitoring only the loss transition of the Cycle-GAN models. As a result, inventory estimation accuracy had to be examined for all fake CG images translated by the models in each training stage, as shown in Fig 9.

The second issue is related to inventory estimation accuracy. As shown in Fig. 10, the MSE obtained using fake CG images was approximately 9.2 times that of using CG images. Furthermore, the transition of MAE varied with increasing nut quantity as shown in Fig. 14. In particular, the MAE increased at 25, 30, and 70 or greater nuts. However, to use this method for stock-taking, it will be necessary to maintain a certain inventory estimation accuracy regardless of the part quantity.

To address these issues, it will be necessary to make the loss function of the Cycle-GAN reflect the loss function of the regression model. Using this model, it is expected that the Cycle-GAN model can be trained to optimal for the regression model to estimate inventory. This will be the focus of future work.

6 CONCLUSION

Inventory estimation of bulk containers using the regression model of deep learning has achieved certain accuracy. However, to apply this method to an actual factory, there were two issues that needed to be addressed, i.e., a large amount of training data must be prepared, and the deterioration of estimation accuracy caused by the domain gap with scene changes must be prevented.

For these problems, in this study, training a regression model using CG images and estimating inventory using fake CG images translated from actual images by Cycle-GAN were investigated. Comparative evaluations of the model's estimation accuracy were performed using CG images, fake CG images, and actual images. As a result, inventory estimation accuracy could be improved using fake images rather than the original images. On the other hand, the estimation accuracy obtained using fake CG images was less than when using CG images.

To improve accuracy when using fake CG images, it will be necessary to reflect the loss of the regression model into the loss function of the Cycle-GAN model, which will be the focus of future work.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 19K11985, and a research grant from Toukai Foundation for Technology.

REFERENCES

[1] A. Anoosheh, et al., “Combogan: Unrestrained scalability for image domain translation,” Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, pp. 783–790 (2018).

[2] A. Bansal, et al., “Recycle-gan: Unsupervised video retargeting,” Proc. European Conference on Computer Vision (ECCV), pp. 119–135 (2018).

[3] A. Cherian, and A. Sullivan, “Sem-GAN: Semantically-consistent image-to-image translation, ’ ’ Proc. 2019 IEEE Winter Conf. on Applications of Computer Vision, pp. 1797–1806 (2019).

[4] F. Chollet, “Deep learning with python,” Manning Publications Co. (2017).

[5] F. Fang, et al., “High-quality nonparallel voice conversion based on cycle-consistent adversarial network,” Proc. 2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 5279–5283 (2018).

[6] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, “Deep learning,” MIT press (2016).

[7] I. Goodfellow, “Nips 2016 tutorial: Generative adversarial networks,” arXiv preprint arXiv:1701.00160 (2016).

[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] P. Isola, et al., “Image-to-image translation with conditional adversarial networks,” Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017).

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

[11] T. Kim, et al., “Learning to discover cross-domain relations with generative adversarial networks,” Proc. 34th Int. Conf. on Machine Learning-Volume 70, pp. 1857–1865 (2017).

[12] A. Krizhevsky, I. Sutskever, G.E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, pp. 1097–1105 (2012).

[13] T. Kudo, and R. Takimoto, “A proposal of deep learning training data generator for image recognition,” Proc. Int. Wokshop on Informatics (IWIN2018), pp. 119–126 (2018), http://www.sist.ac.jp/˜kudo/research/IWIN2018_119_126.pdf (reffered Oct. 19, 2019).

[14] T. Kudo, “CG Utilization for Creation of Regression Model Training Data in Deep Learning,” Procedia Computer Science, Vol. 159, pp. 832–841 (2019).

[15] E. Linder-Norén, “Keras implementations of Generative Adversarial Networks”, https://github.com/eriklindernoren/Keras-GAN (reffered Oct. 19, 2019).

[16] M. Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” Advances in neural information processing systems, pp. 700–708 (2017).

[17] 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).

[18] A. Mordvintsev, C. Olah, M. Tyka, “DeepDream - a code example for visualizing Neural Networks,” Google AI Blog, https://mng.bz/xXlM (reffered June 5, 2019).

[19] 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).

[20] Y. Wen, et al., “A discriminative feature learning approach for deep face recognition,” European conference on computer vision, Springer, pp. 499–515 (2016).

[21] L. Wei, S. Zhang, W. Gao, and Q. Tian, “Person transfer gan to bridge domain gap for person re-identification,” Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 79–88 (2018).

[22] L. Wu, Y. Wang, and L. Shao, “Cycle-consistent deep generative hashing for cross-modal retrieval,” IEEE Transactions on Image Processing, Vol. 28m No. 4, 1602–1612, (2018).

[23] Y. Yuan, et al., “Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks,” Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, pp. 701–710 (2018).

[24] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” Proc. IEEE Int. Conf. on Computer Vision, pp. 2223–2232 (2017).

(Received October 29, 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.