Title: Contour generation for object detection utilizing Cycle-GAN with error monitoring
Author: Tsukasa Kudo

In Proceedings of International Workshop on Informatics (IWIN2021), September 12-13, 2021.
Virtually Obama, Fukui.
http://www.infsoc.org/conference/iwin2021/download/IWIN2021-Proceedings.pdf

This paper is an excerpt from the above proceedings.

Contour Generation for Object Detection Utilizing Cycle-GAN with Error Monitoring

Tsukasa Kudo[†]

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

Abstract - In recent years, with the spread of IoT, a huge amount of image data has been input into systems, and it has become necessary to automatically extract the required information from it. So, various studies on object recognition have been carried out, and remarkable development has been achieved especially by utilizing deep learning. Here, when the target area is small in the image, it is necessary to detect the target object and extract its area for its recognition firstly. In this field as well, various studies utilizing image processing and deep learning are being actively conducted, and improvements in efficiency and accuracy have been achieved. However, to specify the target area or contour in a pixel-to-pixel manner, it is necessary to prepare each pair of the original and its ground truth images as training data, which is a practical obstacle. In this study, I propose a method of translating a target image into a contour-enhanced image using Cycle-Consistent Adversarial Networks (Cycle-GAN), which does not require pairs of training mages. Furthermore, through experiments, it is shown that each contour of the targets can be detected collectively even for plural dense objects.

Keywords: Cycle-GAN, Object detection, Contour detection, Image processing, Image-to-image translation

1 INTRODUCTION

At present, a huge amount of data is input as big data from various sensors with the progress of IoT. And, attempts to automatically extract useful information from these data by utilizing deep learning (DL) are widely made, and high discrimination accuracy is achieved [17], [26]. Among such sensors, to monitor targets by images, a large number of cameras such as surveillance, in-vehicle, river, and wearable cameras have been deployed, and necessary information is automatically extracted from various videos and images.

Here, when the target area in the image is relatively small, it is necessary to detect the target object to extract its area, then the target recognition is performed in this area. For example, in face recognition, a face area is extracted from an image using the Haar-Like features, then face recognition is performed in this area to identify the target person [27].

As such a research field, the author has dealt with the theme of automatically extracting the necessary information for inventory management of the parts shelves in machine assemble factories. Since a factory often has thousands or more inventory shelves, efficient data collection and automatic information extraction are required. So, I have been studying a method of having workers wear wearable cameras and automatically collecting images of inventory shelves during their work. Even with this method, to recognize the target object such as parts, if their areas are small in an image, it is necessary to extract the target area.

For this problem, I focused on the fact that workers pick up parts by hand when they conduct works about inventory operations such as replenishment and shipping. And, using optical flow, which is a representation of the movement of an object between adjacent frames in videos, it was shown that the target object area can be extracted in this case [11]. However, it was a remainder problem to extract the target area for the still target objects, especially from the still image including the plural dense objects, such as the lined-up parts containers in the factory.

Regarding object detection, various methods have been proposed in the conventional image processing field such as area segmentation and contour detection [29]. In addition, in recent years, various methods using DL are proposed [16], [28], and it has been shown that the target area or contour can be extracted in a pixel-by-pixel manner with some methods. However, since these methods need each pair of the original and its ground truth images as training data, it is an obstacle to actual application.

On the other hand, Cycle-Consistent Adversarial Networks (Cycle-GAN) have been proposed, which is a kind of generative adversarial networks (GAN) [32], [5]. And, it has been shown that mutual image translation between two different types of images can be performed by Cycle-GAN such as between photographs of horses and zebras. In addition, just both types of images are necessary for the training data, and they do not need to be paired. This suggests that an image including target objects can be translated into one that emphasizes the contour of the objects using the training data prepared efficiently without making data pairs as above-mentioned, and the target object region can be extracted from it.

In this study, a method to extract contours using a contour extraction model (CE-model) is proposed, which is based on Cycle-GAN. In this method, the following two types of images are translated mutually. One is the original image; another is its contour emphasis image, in which the contour area is represented as so brighter, and other areas are darkened. And, it is shown that the contour of the target object can be obtained from this contour emphasis image generated by the CE-model trained with these images.

In Cycle-GAN, the accuracy of the extracted contour cannot be directly monitored because supervised training using ground truth is not performed. In addition, it has been found that the loss of CyckleGAN and the error of the object created by it does not always consistent, through my previous re-

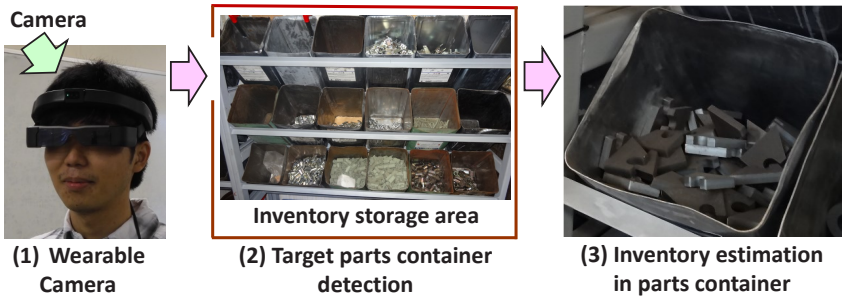

Figure 1: Automatic inventory management using videos

search [12]. So, this method incorporates a function of monitoring this error to stop the Cycle-GAN model training at the optimum timing. Furthermore, through experiments targeting books arranged on bookshelves, it is shown that targets' contours can be extracted collectively from a still image even including plural dense objects by this method.

The remainder of this paper is organized as follows. Section 2 describes related works and motivation for this study, and Sec. 3 proposes a contour extraction method. Section 4 shows the implementation of the experimental system and evaluations, and its results are discussed in Sec. 5. Finally, Sec. 6 concludes this paper.

2 RELATED WORKS AND MIOTIVATION

Nowadays, with the progress of IoT, it has become possible to connect various cameras to the Internet and easily share and analyze recorded videos. As a result, using cameras such as wearable cameras and in-vehicle cameras, it has become possible to automatically extract necessary information from the scenery we see casually in our daily lives and utilize it efficiently.

Regarding such IoT applications, the author has been working on a study to automate inventory management in machine assembly factories. There are various parts stored in the bulk container shown in Fig. 1 (3), and they cannot be counted visually from the outside. Furthermore, since the number of these containers often reaches several thousand or more, efficient inventory management has been an issue. For example, an attempt to grasp quantities of parts by measuring the weight of each container has been unsuccessful, because the uneven distribution of parts in containers causes a non-negligible error and too many scales were required. For this issue, I focused on the fact that inventory fluctuates only with inventory operations such as the worker's replenishing or shipping of parts, and conceived to extract the necessary inventory information automatically from the videos shot by the wearable camera worn by the worker.

Using smart glasses equipped with the camera shown in Fig. 1 (1), workers can take videos of the target object without extra load and confirm the shot video by displaying it on the see-through glasses. And, I have shown that some necessary information for inventory management could be extracted automatically by applying the multi-class classification and regression model of DL to these videos in my previous study [9], [10]. By the former, the current position such as entering the inventory storage area shown in Fig. 1 (2) could be determined; by the latter, inventory quantities of parts in the bulk containers shown in (3) could be estimated, even though it could not be counted from the outside visually as abovementioned.

Furthermore, I have shown even when a target object's area was relatively small in an image, it could be extracted by utilizing the optical flow when the target is moving against the background, such as the worker picking up parts during the work [11]. Supplementally, in this case, the target area has to be extracted first to maintain its recognition accuracy, similar to such as face detection in face recognition. However, extraction of a target area from dense still objects, such as the lined-up bulk containers as shown in Fig. 1 (2) remained an issue. And, this is the subject of this study.

To detect such objects, some methods have been proposed in the conventional image processing fields. In contour tracking, the target's contour is extracted to identify its area; In edge detection, the boundary of the target is specified using a filter or the like; In segmentation using such as mean-shift clustering, pixel value or texture gradient is utilized to determine the target area [29]. But, it has been pointed out that these contour detection methods using image processing become difficult tasks when contours are incomplete or not closed [4].

On the other hand, with the progress of DL in recent years, studies on object detection have been actively conducted, and various methods have been proposed. One is based on the region proposal that detects the bounding boxes where the objects exist, and YOLO (You Only Look Once) achieved high efficiency by detecting images by CNN (Convolutional Neural Network) processing only once; SSD (Single Shot Detector) enabled to detect objects of various sizes with one processing, and RatinaNet improved its efficiency [16], [19], [18], [14], [15]. Furthermore, even for dense objects, some ways are shown to detect the individual object's region, such as detecting new objects repeatedly (IterDet) and separating duplicate regions by post-processing [1], [20]. However, since these region proposals use bounding boxes, namely rectangles, the exact area of targets cannot be specified.

To detect object regions or contours in a pixel-by-pixel manner, various approaches have been studied. A basic method has been proposed in which each pixel is judged whether in an object's contour using CNN [23]. For the methods based

on the region proposal, semantic and instance segmentation have been proposed, which are segmentation for each object's class and object itself respectively [31], [7]. Furthermore, after the method of image-to-image translation with cGANs (conditional GANs) between original and feature images was shown as pix2pix [8], methods using GAN have been studied actively [13], [24], [28], [30]. However, since these methods require pairs of the original and ground truth images as training data, there is an obstacle to their practical use.

Cycle-GAN has been proposed as one of the GANs for image-to-image translation. By using it, images in a domain can be translated to the ones in another domain mutually, and the original paper showed examples of mutual translation between photographs and painter's drawings, summer and winter photographs, and so on [32]. Regarding the above-mentioned obstacle for practical use, Cycle-GAN has an important feature that it is not necessary to prepare each image pair of two domains as training data, that is, it is easier to prepare training data.

In my previous study, I have utilized Cycle-GAN to prepare efficiently the training data for the model to estimate the inventory quantity shown in Fig. 1 (3). Concretely, this model was trained with CG images generated automatically, and inventory was estimated with fake CG images translated from photographs using Cycle-GAN. As a result, it was shown that the estimation accuracy could be improved compared to the case of using the original photographs [12]. However, through this study, it was also found that there was no correlation between the loss of Cycle-GAN in training and the transition of the above-mentioned estimation accuracy, that is, there was an issue when to stop model training.

Similarly, various applications of Cycle-GAN have been proposed. The first is the augmentation of training data, which is used in the fields where it is difficult to generate sufficient training data for DL, such as medical treatment and detection of plant lesions [22], [25]. The second is to translate the original image into an image that is easier to detect objects as a preprocessing, and methods combined with such as YOLO and RetinaNet have been proposed [21], [3]. However, I could not find the study targeting contour detection for still images including dense objects.

The goal of this study is to develop a method to extract the target objects' contours collectively from a still image including plural dense objects such as the lined-up containers' image shown in Fig. 1 (2) using Cycle-GAN. And, its results are assumed to be used for target detection and recognition. For example, by identifying the inventory shelf currently in operation, the target part can be identified. And, its inventory quantity can be estimated using the above-mentioned regression model of my previous study. In addition, the motivation of this study is the question of whether the contours of the target objects can be extracted from contour emphasis images which are translated from original images using Cycle-GAN.

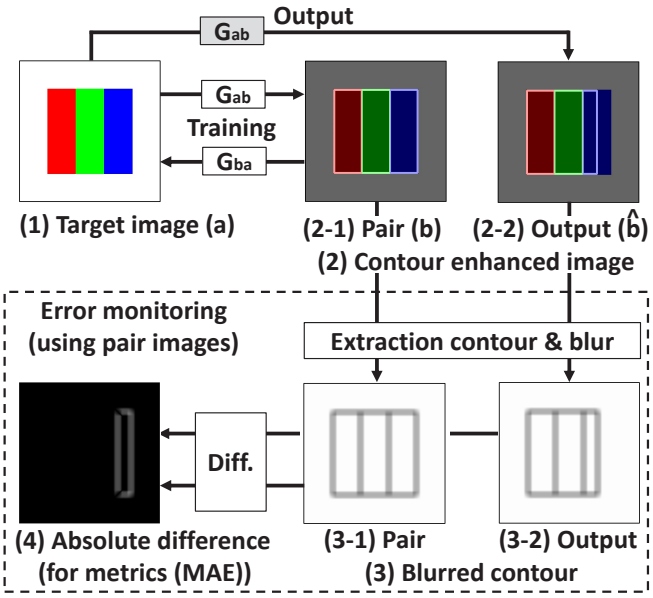

Figure 2: Contour extraction method utilizing Cycle-GAN

3 PROPOSAL OF CONTOUR EXTRACTION MODEL

3.1 Contour Generation Applying Cycle-GAN

Figure 2 shows the method of extracting the contours of the target's object from a still image including plural dense objects using Cycle-GAN. Figure 2 (1) shows the image that includes three dense objects. (2-1) is an image that emphasizes the contour of the target's objects in (1). In this study, CE-model utilizing Cycle-GAN is proposed, which is trained with images in these two domains (1) and (2-1). From images in the domain (1), its corresponding contour emphasis images shown in (2-2) are generated using the CE-model.

When the image of Fig. 2 (1) is shown by a and the image of (2-1) is shown by b, generators of the CE-model that perform mutual translation between them are shown below.

$$\hat{a} = G_{ba}(b) \tag{1}$$

$$\hat{b} = G_{ab}(a) \tag{2}$$

Here, $\hat{a}$ and $\hat{b}$ are fake images of the image a and b respectively. For image b, the CE-model is trained using discriminators that monitor the following three losses same as Cycle-GAN.

$$L_b = \| G_{ba}(b) - a \| \tag{3}$$

$$L_c = \| G_{ab}(G_{ba}(b)) - b \| \tag{4}$$

$$L_i = \| G_{ab}(b) - b \| \tag{5}$$

Here, $\|\|$ indicates an error, and similar losses are monitored for a. L_b evaluates the error between the fake image $\hat{a}$ and a; L_c evaluates the reconstruction image generated by applying these two generators sequentially, namely between the fake image of b and b itself; L_i evaluates the identity image, which is generated by the generator for this image itself. In training, these losses are added with a specified weight to make the total loss.

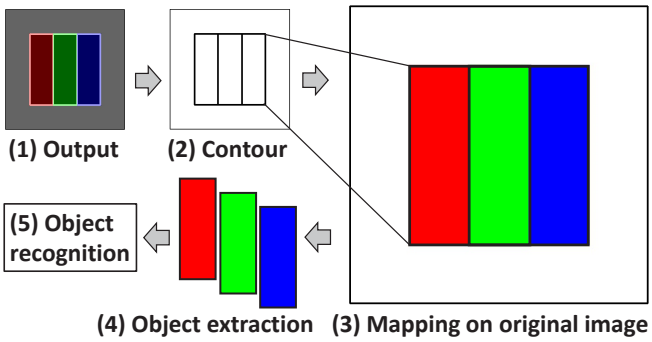

Figure 3: Object detection from contour emphasis image

As mentioned in Sec. 2, in Cycle-Gan training, ground truth data are not used, that is, the error of the extracted contour cannot be monitored. Also, its loss and the error of the object created by it do not always consistent. So, in this method, the difference between the contours, namely their error, shown in Fig. 2 (2-1) and (2-2) is monitored as metrics. The former is the contour emphasized data b, which is added as ground truth and made from target image (1) by emphasizing the contour of each object; The latter is generated from the corresponding image a by the generator G_{ab}, namely the fake image of b. The training ends when the difference of contours generated from each of them becomes the minimum. Note that such ground truth is not necessary for Cycle-GAN training itself.

I show the error monitoring feature of the CE-model in the dashed rectangle of Fig. 2. The contours of (2-1) and (2-2) are extracted, and blur is applied as shown in (3-1) and (3-2), and the absolute difference between both is made as shown in (4). Lastly, the mean absolute error (MAE) between them is calculated, which is the average brightness of the image shown in (4). This MAE is used for the metrics. Here, blur is for reflecting the distance between both contours into the metrics. For example, when the contours are extremely close, MAE is small and increases as the distance increases.

3.2 Object Detection Using Contour Emphasis Image

Figure 3 shows the flow of object detection using the contour emphasis image that is the output of the CE-model. Figure 3 (1) shows the output corresponding to Fig. 2 (2-2). The contour image shown in Fig. 3 (2) is constructed from the image shown in (1) by the procedure of converting this image to grayscale and extracting the area with brightness above the specified threshold as the contour area.

The image in Fig. 3 (2) is reduced because it is the output of the CE-model. Therefore, the contour of each object of this image is converted to its original scale, then mapped to the original image as shown in (3). Based on this contour, as shown in (4), each target object area is extracted. After that, object recognition is performed by the method according to the application such as template matching, multi-class classification of DL, or character recognition.

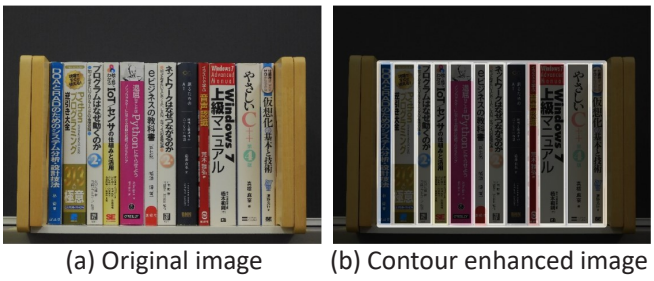

Figure 4: Target image of experiment: books on bookshelf

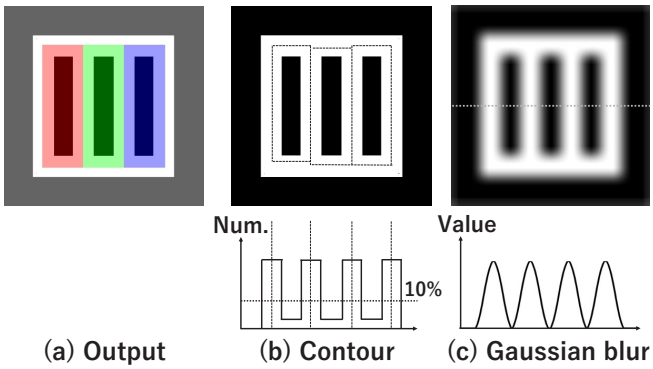

Figure 5: Contour extraction and blur for metrics

4 IMPLEMENTATION AND EVALUATIONS

4.1 Configuration of Experimental System

To evaluate the effectiveness of the proposed method, I constructed an experimental system for the target's contour extraction and conducted experiments. The books arranged on the bookshelf as shown in Fig. 4 were used for the experimental targets as still images including plural dense objects. Figure 4 (a) shows the original image, and (b) shows the image that emphasizes the contour. Each image size was 691×518 pixels; the brightness of each RBG color channel was in the range [0, 255]; all the contour was set manually. For (b), to emphasize the contour, the brightness was mapped based on the threshold 128 with an error of 28. That is, the contour area was mapped into the range $[156, 255]$, and another area into $[0, 99]$.

The experiment was carried out on a personal computer, which CPU was i9-10850K (3.6 GHz), memory was 64 GB, and GPU was GeForce RTX 3090 with 24 GB memory, and OS was Windows 10. Tools and programming languages were Keras Ver. 2.4.3, Tensorflow-GPU Ver. 2.4.1, and Python Ver. 3.7.10. Code of the CE-model was created based on the published Cycle-GAN code [2], along with adding the necessary functions to experiment, such as calculating the metrics, and saving and displaying results.

For the CE-model training, the above bookshelf images were resized to 128×128, and the batch size was 32. For the configuration of this model, residual networks (ResNet) were used, which have shortcut connections for skipping one or more layers to increase the depth of the network [6]. In addition, the weights of the losses in Eqs. (3) to (5) were set

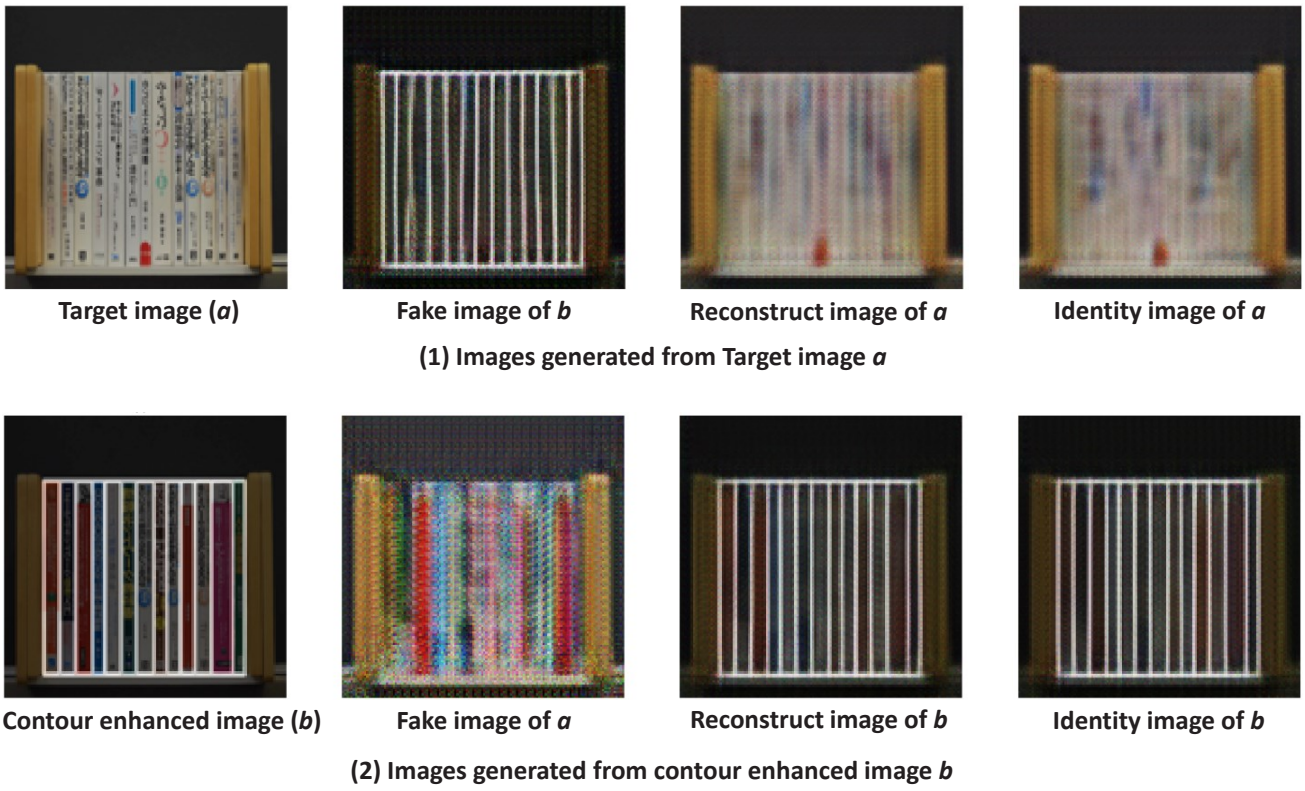

Figure 6: Example of generated images using CE-model

to 4, 10, and 2, and their weighted summation was reflected for the training.

Figure 5 shows the implementation to extract the contour shown in Fig. 3 (2) and to blur it for calculating the metrics as shown in Fig. 2 (3). The image shown in Fig. 5 (a), which was the output image of the CE-model shown in Fig. 2 (2-2), was converted to grayscale. Then, the contour and another area were separated based on a threshold, which was set to 111 in consideration of the grayscale error. As a result, the contour area was set to white, and another area was set to black as shown in Fig. 5 (b). Then, Gaussian blur with kernel size 5×5 and standard deviation of 0.3 was applied as mentioned in Sec. 3.1, and the image shown in Fig. 5 (c) was created.

To calculate the metrics, pair of blurred images were used. One was made from the target's contour emphasized image as shown in Fig. 2 (3-1); another was made from the output image as shown (3-2). Then, the MAE of the absolute difference between both, namely metrics, was calculated by Eq. 6.

$$m_e = \sum_{i=1}^{n} \sum_{j=1}^{n} |x_{eij} - b_{ij}| / n^2 \tag{6}$$

Here, b_{ij} and x_{eij} show brightnesses in pixels of the former and latter respectively; e is the epoch number; i and j indicate the pixel coordinates (i, j), and n is the number of pixels in each coordinate axis. In the training, m_e was calculated for each epoch. Every time m_e became the minimum compared before, the model weight was saved as the best weight. Then, when m_e did not improve the specified number of times, the training was completed. In this experiment, this number was set to 15 epochs.

Contour extraction was performed on the image in Fig. 5 (b), and firstly the vertical contour was detected. As the procedure, as shown in the lower graph of (b) was created, which indicated the number of pixels in the contour area in the vertical direction for each horizontal position. And, the horizontal area, of which the contour pixel number was 10 % or more was selected as a contour candidate area. In the case of Fig. 5, there were 4 candidate areas, and for each area, the position of the largest number of pixels was selected as the vertical part of the contour, which is indicated by the broken line.

Next, the number of pixels in the contour area of the horizontal direction was counted between each adjacent vertical part of the contour, and the vertical area with 50 % or more numbers was selected as the horizontal part candidates of the contour. And, similar to the vertical part, the horizontal part of the contour was selected. Then, the contour of each object was extracted as shown by the broken line in Fig. 5 (b). In this experiment, they were rectangles.

4.2 Evaluations of Contour Extraction

CE-model was trained using the training data, then the accuracy of contour extraction was evaluated. For training and evaluation, 128 pairs of the original image of the bookshelf and the contour emphasis image shown in Fig. 4 were prepared and inflated to 256 pairs by horizontal flip. They were divided into 204 pairs of training data and 52 pairs of test data for monitoring metrics m_e of Eq. 6. Data were randomly selected and shuffled in each batch of training, without consid-

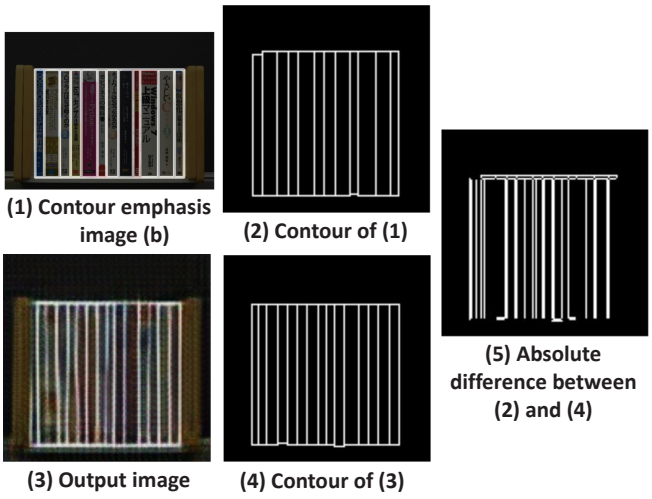

Figure 7: Error of extracted contour

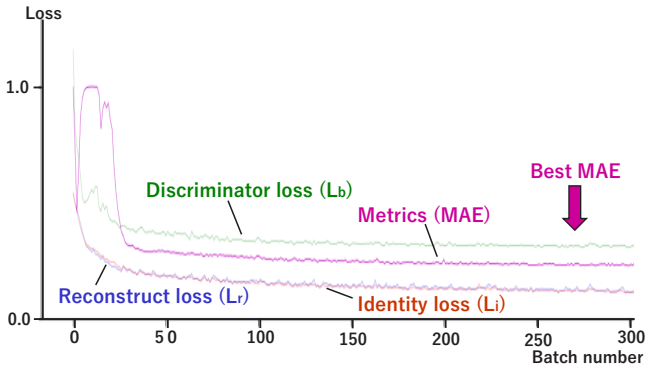

Figure 8: Transition of losses and metrics with training

ering the pair. In this experiment, the best metrics 0.2278 was obtained at epoch 54.

Figure 6 shows examples of images generated by the training data at the above-mentioned epoch 54 during the training. Similar to Fig. 2, the target image, namely the books on the bookshelf, is shown by a, and the contour emphasis image is shown by b. The upper row shows the images generated from a. From the left, the original image a, the fake image represented by $G_{ab}(a)$, the reconstructed image $G_{ba}(G_{ab}(a))$, and the Identity image $G_{ba}(a)$ are shown. Similarly, the lower row shows images generated from b. Since the pairs were not maintained as above-mentioned, the books are different from the upper row.

As a result, as shown in the second image from the left of Fig 6 (1), clearly emphasized contours were generated from the target image. However, as shown in the right two images in (1), the clear characters in the book's back cover could not be generated in the output of the CE-model.

A contour emphasis image b of the test data and the contour extracted from it are shown in (1) and (2) of Fig. 7. Similarly, a fake contour emphasis image and the contour are shown in (3) and (4), which were generated from the target image a corresponding to the above-mentioned b using the CE-model after training. The aspect ratio of (1) is different because it is before resizing (691×518) and the others are output images

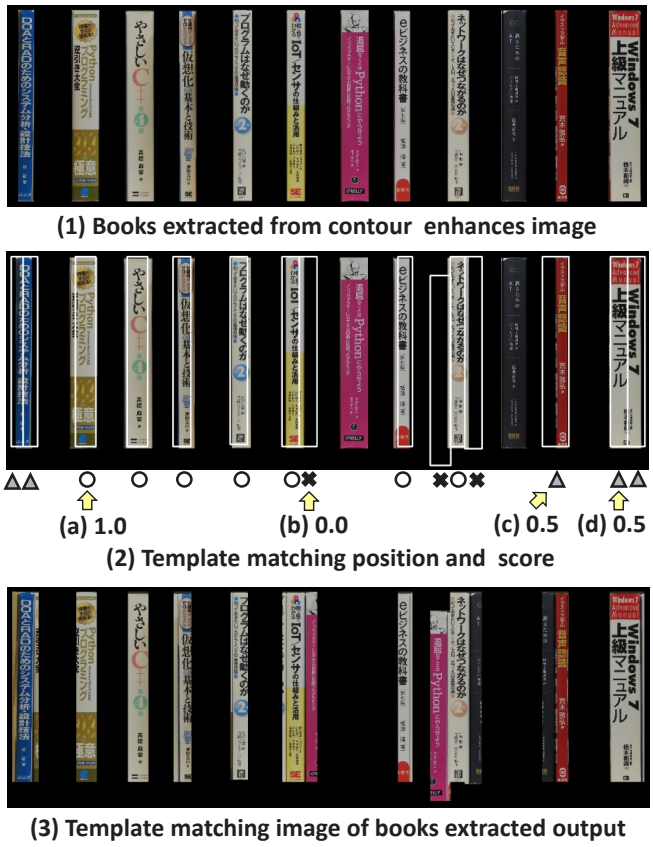

Figure 9: Book recognition results using extracted contours

from the CE-model (128×128). Since there was a difference between contours (2) and (4), the absolute difference between them was created as shown in (5). The thin lines indicate that their corresponding contours are in different positions, and the thick lines show the contours adjacent to each other though they are in different positions.

In summary, contours could be extracted using the CE-model, but there were some errors in contour position.

Figure 8 shows the transition of each loss and metrics of the CE-model during training. The horizontal axis of the figure shows the batch number, and 1 epoch corresponds to 5 batches. During the period immediately after the start of training, each loss and metrics showed an individual tendency, but after 30 batches, namely 6 epochs, they showed the same tendency. The arrow indicates epoch 54 (batch number 270) when the metrics (MAE) became the best, namely the minimum.

4.3 Evaluations of Object Recognition

To evaluate the effect of contour extraction error described in Sec. 4.2 on object recognition, all test data excluding flipped images were evaluated by template matching. Concretely, for each test data, a pair of contours shown in (2) and (4) of Fig. 7 was used, and each book image was extracted from the original image shown in (a) of Fig. 4 by the procedure shown in Fig. 3.

Figure 9 (1) shows each book image extracted from the original image using the contour shown in Fig. 7 (2), namely contour of the contour emphasis image. Similarly, each book

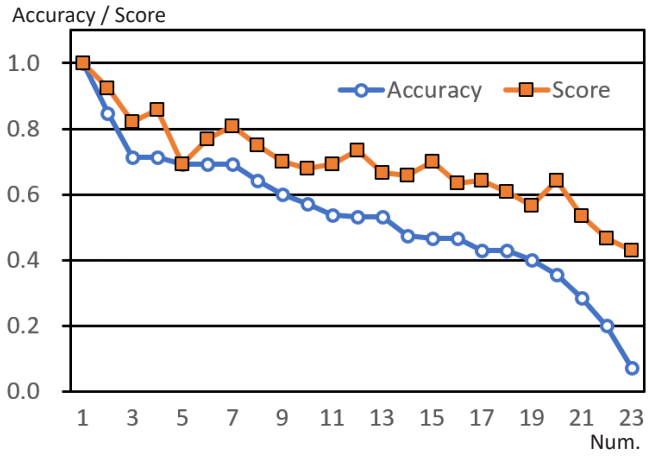

Figure 10: Variation in book recognition accuracy

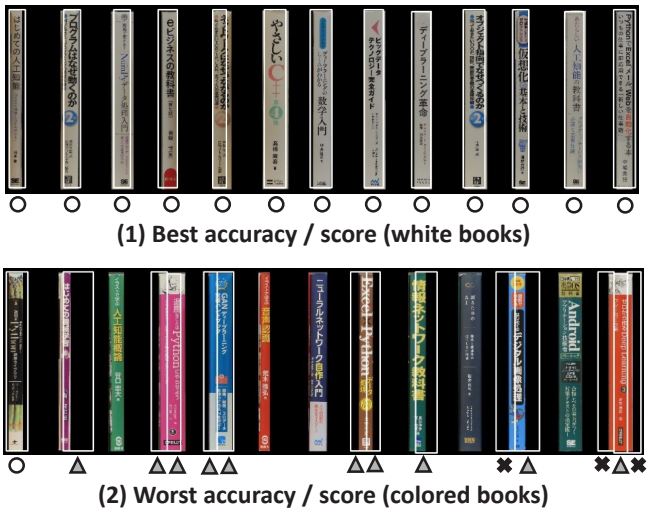

Figure 11: Best and worst results of book recognition using extracted contou

image was extracted using the contour shown in Fig. 7 (4), namely contours based on the output image of the CE-model. Next, using each latter book image, the corresponding book in the image shown in Fig. 9 (1) was searched using template matching, in which the normalized correlation coefficient matching method was used. (2) shows the matching position of each book by a white rectangle; (3) shows each book image based on the extracted contour shown in Fig. 7 (4).

The following three scores were introduced according to the matching level to evaluate the accuracy of object recognition, as shown in Fig. 9 (2). (a) shows the case of almost the same indicated by ◯, which score is 1.0; (b) shows the case of almost the difference indicated by ×, which score is 0.0; (c) and (d) shows the case between the both indicated by △: there is an extra area or only a part area respectively whereas the same. Note that (c) and (d) are the case when 50% or more of the target book is included, and (b) is the case of less than 50%.

The percentage of the extracted contours with a score of 1.0 (hereinafter, estimation accuracy), and the average score were calculated. In the case of Fig. 9, the number of extracted contours was 15, 1.0 was 7, 0.5 was 5, 0.0 was 3, so the estimation accuracy became $7/15 = 0.47$ and the average score $9.5/15 = 0.63$. Figure 10 shows a graph in which the estimation accuracy and average scores are arranged in descending order of estimation accuracy for all test data excluding flipped images. Both varied widely depending on the type of book, with averages of estimation accuracies 0.537 and of scores 0.694.

The image with the best and worst accuracy are shown in Fig. 11 (1) and (2) respectively. As shown in (1), the best accuracy was obtained when only white books were targeted. On the contrary, as shown in (2), the accuracy of colored books, especially dark-colored books, tended to deteriorate. This tendency was the same in the case shown in Fig. 9, and the darkest-colored book could not be detected. Conversely, white books were relatively well detected.

5 DISCUSSIONS

The advantage of this method is that it is not necessary to prepare pairs of the original and ground truth images. In this experiment, to monitor the metrics, these pairs were prepared, namely original and contour-emphasized images. However, unlike my previous study mentioned in Sec. 2, for the CE-model in this study, as shown in Fig. 8, the transition of the metrics and CE-model's losses indicated the same tendency. The reason is considered that both are dealing with the image itself. That is, the ratio of metrics data can be reduced. Therefore, for uniform objects such as books on a bookshelf, it is expected to generate training images efficiently. For example, the contour-enhanced image can be divided as shown in Fig 9 (1) and rearranged for augmentations.

In this study, the method to extract contours of the plural dense objects from the still image was proposed, which is based on Cycle-GAN to translate an original image into a contour emphasis image. And, the experiments targeting books on the bookshelf were conducted. As a result, it was shown that their contours could be extracted and each book could be recognized using these contours as shown in Figs. 6 and 9. In addition, it was shown that plural contours could be extracted collectively as shown in Fig. 9.

However, as shown in Fig. 10, the accuracy of extracted contour was dispersed depending on the color of the target book. As shown in Fig. 11, when the book's back cover was white, the contour was extracted almost correctly. On the other hand, the accuracy deteriorated for a dark-colored book. Its reason is considered that the shadow of the boundary of the book greatly contributes to extracting contour, and this improvement is a future challenge.

The advantage of contour detection is considered that it is not necessary to consider the size of the target object for object detection. In general, since the size of the target is unknown in object detection, the approach is adopted, in which several sizes of the template are prepared and applied sequentially. On the other hand, in this method, since the size of the target can be grasped from the extracted contour, for exam-

ple, it is possible to change the size of the template itself in advance of template matching such as shown in Fig. 9.

6 CONCLUSIONS

When recognizing small objects in a still image, it is necessary to detect the object first. However, to detect each object in a pixel-by-pixel manner, pairs of the original and ground truth images had to be prepared as training data in the conventional methods using DL, and it caused an obstacle to their practical use.

In this study, I proposed an object detection method based on the contours, in which a contour emphasis image was generated from the original image by the model applying CycleGAN, namely the CE-model. This model is trained mutually using the original and contour emphasis images, and both do not need to associate as such pairs. Furthermore, through the experiments targeting books arranged on a bookshelf, it was shown that it was possible to collectively extract contours and recognize each object using them even from dense objects in a still image.

Future studies will focus on contour extraction accuracy improvement of this model, and its effectiveness evaluations for objects of various shapes.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 19K11985.

REFERENCES

[1] X., Chu, A. Zheng, X. Zhang, and J. Sun, "Detection in crowded scenes: One proposal, multiple predictions," Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition, pp. 12214–12223 (2020).

[2] D. Foster, "Generative deep learning: teaching machines to paint, write, compose, and play," O'Reilly Media (2019), https://github.com/davidADSP/GDL_code (referred May 24, 2021).

[3] P. Gao, T. Tian, L. Li, J. Ma., and J. Tian, "DE-CycleGAN: An object enhancement network for weak vehicle detection in satellite images," IEEE J. Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 14, pp. 3403–3414 (2021).

[4] X.Y. Gong, H. Su, D. Xu, et al. ,"An overview of contour detection approaches," Int. J. Autom. Comput., Vol. 15, No. 6, pp. 656–672 (2018).

[5] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, "Generative adversarial networks," Communications of the ACM, Vo. 63, No. 11, pp.139–144 (2014).

[6] K. He, X, Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Conf. Computer Vision and Pattern Pecognition, pp. 770–778 (2016).

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," Proc. IEEE Int. Conf. Computer Vision, pp. 2961–2969 (2017).

[8] P. Isola, J.Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1125–1134 (2017).

[9] T. Kudo, and R. Takimoto, "CG utilization for creation of regression model training data in deep learning." Procedia Computer Science, Vol. 159, pp. 832–841 (2019).

[10] T. Kudo, "A proposal for article management method using wearable camera," Procedia Computer Science Vol. 178, pp. 1338–1347 (2020).

[11] T. Kudo, "Moving Object Detection Method for Moving Cameras Using Frames Subtraction Corrected by Optical Flow," Int. J. Informatics Society, Vol. 13, No. 2, pp. 79–91 (2021).

[12] T. Kudo, "CG training model application method using cycle-consistent adversarial network," Int. J. Informatics Society, Vol. 12, No. 1, pp.41–48 (2020).

[13] M. Li, Z. Lin, R. Mech, E. Yumer, and D. Ramanan, "Photo-sketching: Inferring contour drawings from images," 2019 IEEE Winter Conf. Applications of Computer Vision (WACV), pp. 1403–1412 (2019).

[14] T.Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2117–2125 (2017).

[15] T.Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection." Proc. IEEE Int. Conf. Computer Vision, pp. 2980–2988 (2017).

[16] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikáinen, "Deep learning for generic object detection: A survey," Int. J. Computer Vision, Vol. 128, No. 2, pp. 261–318 (2020).

[17] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," IEEE Communications Surveys & Tutorials, Vol. 20, No. 4, pp. 2923-2960 (2018).

[18] M. Rajiput, "YOLOv5 is here! elephant detector training using custom dataset & YOLOV5," https://towardsdatascience.com/yolo-v5-is-here-b668ce2a4908 (referred May 24, 2021).

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 779–788 (2016).

[20] D. Rukhovich, K. Sofiiuk, D. Galeev,O. Barinova, and A. Konushin, "IterDet: Iterative Scheme for Object Detection in Crowded Environments," arXiv preprint arXiv:2005.05708 (2020).

[21] K. Saleh, A. Abobakr, M. Attia, J. Iskander, D. Nahavandi, M. Hossny, and S. Nahvandi, "Domain adaptation for vehicle detection from bird's eye view LiDAR point cloud data," Proc. IEEE/CVF Int. Conf. Computer Vision Workshops, pp. 3235–3242 (2019).

[22] V. Sandfort, K. Yan, P. J. Pickhardt, and R.M. Summers, "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks," Scientific reports, Vol. 9. No. 1, pp. 1–9 (2019).

[23] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 3982–3991 (2015).

[24] A. Sindel, A. Maier, and V. Christlein, "Art2Contour: Salient Contour Detection in Artworks Using Generative Adversarial Networks," 2020 IEEE Int. Conf. Image Processing (ICIP), pp. 788–792 (2020).

[25] Y. Tian, G. Yang, Z. Wang, E. Li, and Z. Liang, "Detection of apple lesions in orchards based on deep learning methods of cyclegan and YOLOv3-dense," J. Sensors, Vol. 2019, Article 7630926 (2019).

[26] 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).

[27] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proc. 2001 IEEE Computer Society Conf. Computer Vision and Pattern Recognition, Vol. 1, pp.511–518 (2001).

[28] H. Yang, Y. Li, X. Yan, and F. Cao, "ContourGAN: Image contour detection with generative adversarial network, Knowledge-Based Systems, Vol. 164, pp. 21–28 (2019).

[29] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," ACM Computing Surveys (CSUR), Vol. 38, No. 4, Article 13 (2006).

[30] Z. Zeng, Y. K. Yu, and K. H. Wong, "Adversarial network for edge detection," Int. Conf. Informatics, Electronics & Vision (ICIEV), pp. 19-23 (2018).

[31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2881–2890 (2017).

[32] 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. Computer Vision, pp. 2223–2232 (2017).