Assembly Progress Estimation using Instance Segmentation with Data Augmentation considering Annotation Data

Hayami Shotsu*[a], Shigeki Yumoto[a], Kazuma Miura[a], Sarthak Pathak[b], Alessandro Moro[c], and Kazunori Umeda[b]

[a]Precision Engineering Course, Graduate School of Science and Engineering, Chuo University, Tokyo, Japan
[b]Department of Precision Engineering, Chuo University, Tokyo, Japan
[c]RITECS, Tokyo, Japan

ABSTRACT

In this study, we propose a method for estimating product assembly progress in a factory using instance segmentation. The method detects parts by instance segmentation and estimates the progress based on the number and types of detected parts. The background affects the accuracy of estimation, and the training data is limited. Therefore, we apply data augmentation using random masking to reduce the effect of background information and improve part detection accuracy. Experiments are conducted on a shelf simulating a factory product to confirm the effectiveness of the proposed method.

Keywords: Factory, Deep Learning, Instance Segmentation, Data Augmentation

1. INTRODUCTION

In recent years, advances in AI, IoT, and image processing technologies have driven the development of smart factories.[1] Automating product assembly progress management is particularly important for improving operational efficiency and identifying work-related issues. However, in factories that produce a wide variety of products in small quantities, assembly is primarily performed by human workers. Since manual progress management is time-consuming, work progress is not effectively shared, making it difficult to identify delays and meet deadlines.

To address this issue, an automated method for estimating assembly progress using images captured by fixed-point cameras installed in factories is considered effective.[2] One such method estimates assembly progress based on workers' hand movements.[2] However, this approach becomes impractical when assembling large objects, as assembly motions vary depending on the operator. Another approach involves classifying work processes into predefined classes based on images captured by cameras[3].[4] However, a major drawback of this method is the occurrence of misclassification, as visually similar processes that differ only slightly can be difficult to distinguish.

Therefore, we proposed a progress estimation method utilizing instance segmentation. Instance segmentation simultaneously performs object detection and region segmentation, estimating progress based on the type and number of detected parts. However, in real-world applications, collecting sufficient training data is challenging, and segmentation accuracy is often degraded due to background clutter and occlusion. Data augmentation is a technique for improving accuracy with limited training data. Existing augmentation methods, such as Cutout,[5] Random Erasing,[6] and GridMask,[7] enhance training by masking portions of the input data. However, in product assembly, these methods pose a problem: when parts are completely occluded during training, detection accuracy decreases.

In this study, we propose an assembly progress estimation method that utilizes data augmentation based on annotation data. To improve detection accuracy, relatively small masks are randomly applied to training data, increasing the amount of training data while ensuring that key parts remain visible. In this process, annotation data is used to control the proportion of parts that are occluded within rectangular regions.

*Email:shotsu@sensor.mech.chuo-u.ac.jp

2. RELATED WORK

Oshima et al.[3] proposed a progress estimation method based on class classification using deep learning. First, the product states in the assembly process are defined. The assembly process is divided into an arbitrary number of stages, and the product state at each stage is specified. Next, product images from divided assembly stages are used to train ResNet, a type of convolutional neural network. The trained ResNet model is then used to estimate the assembly process by analyzing input product images. However, a major issue with this approach is the occurrence of misclassification, as assembly stages that are close to each other often have only minor visual differences, making them difficult to distinguish.

Kitsukawa et al.[4] proposed a progress estimation method based on class classification using deep distance learning. Deep distance learning is a technique that transforms input data into a feature space where classes are separated based on the distance or similarity between samples. Unlike general class classification using deep learning, this method intentionally increases the distance between samples of different classes while reducing the distance between samples of the same class. This approach enables accurate inference even with a small amount of training data. However, because inference is performed based on predefined assembly processes, it becomes difficult to make accurate predictions when the assembly order differs from the expected sequence. Additionally, limited training data can further reduce the accuracy of the model.

3. PROPOSED METHOD

3.1 Outline of Proposed Method

The flow of the proposed method is shown in Fig. 1. First, the product state in the assembly process is defined. The assembly process is divided into an arbitrary number of processes, and the product state of each assembly process is defined. Specifically, the type and number of parts installed in the assembly process are set. Next, the model is trained with images of the parts. Next, we train the model on the part images, applying data augmentation. During recognition, we input the images into the learned model and output the detection results of the parts. Progress is estimated from the detection results.

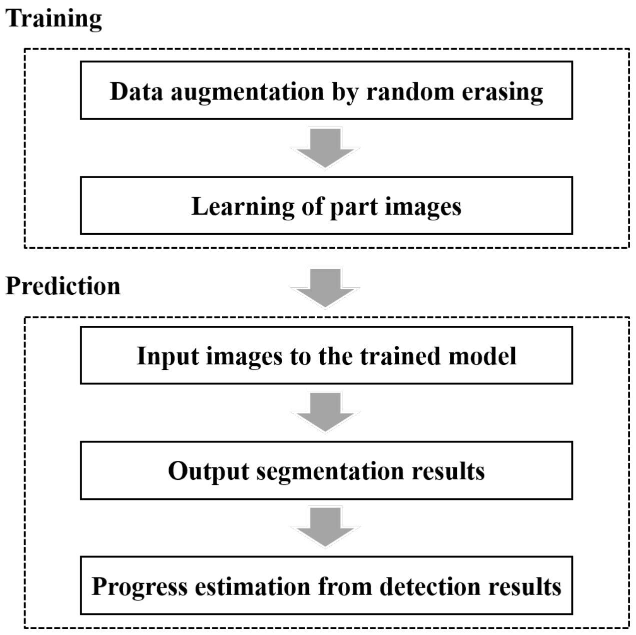

Training
Data augmentation by random erasing
Learning of part images
Prediction
Input images to the trained model
Output segmentation results
Progress estimation from detection results

Figure 1. Flow of the proposed method

3.2 Instance Segmentation

Instance segmentation is a task that performs both object detection and region segmentation. YOLACT[8] is used for instance segmentation. YOLACT is a model that enables fast instance segmentation, and because it is a one-stage model that simultaneously performs object detection and identification, unlike the two-stage Mask R-CNN,[9] it can detect objects in real time.

3.3 Progress Estimation Method

The assembly progress is estimated using the detection results of instance segmentation. If a combination of parts detected in the assembly progress estimation does not fit any of the assembly processes, it is assumed to be an error, as an incorrect assembly process. Unlike previous methods based on class classification,[4] this system can detect abnormalities using only normal images. Fig. 2 shows an example of the proposed method.

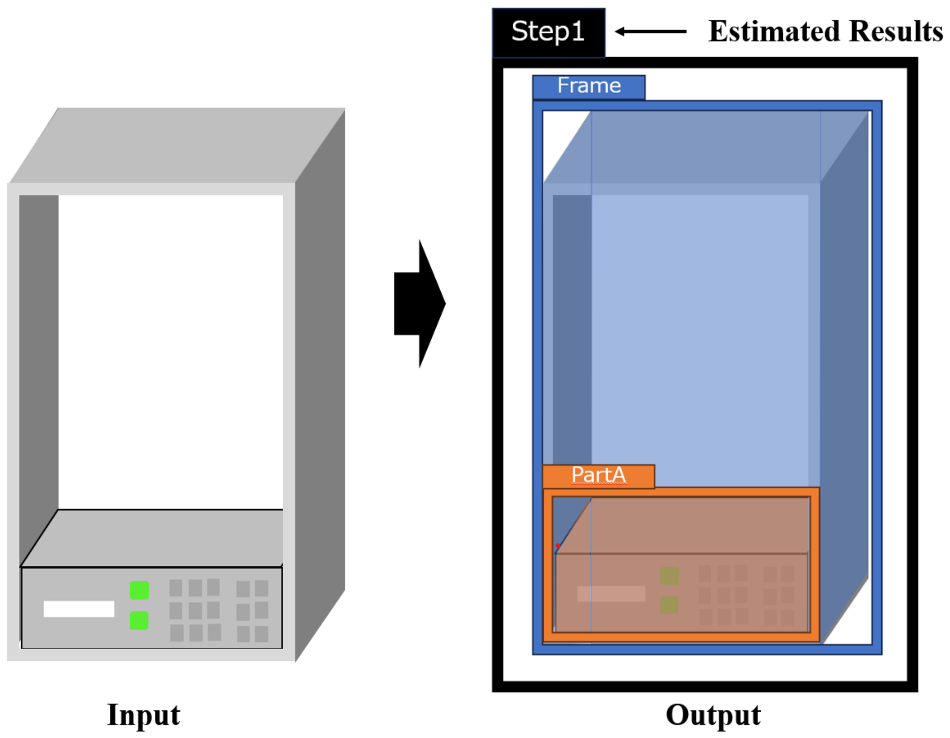


Figure 2. Example of progress estimation

3.4 Data Augmentation

Our problem setting is a factory with several occlusions, and training data availability is low. There are several data augmentation methods to deal with occlusion: Random Erasing[6] masks the image with randomly sized rectangles, and Gridmask[7] masks with regularly arranged squares. However, they hide the target and reduce detection accuracy. Therefore, we propose data augmentation with multiple small random color masks that consider the annotation data so as not to hide the parts completely. Fig. 3 shows an example of annotation data.

Our proposed method uses annotated data to control the rate of parts that are masked by rectangular regions. The occlusion algorithm utilizes Random Erasing.[6] First, the rectangular region's area S_e is set within the range of s_l to s_h. Next, the aspect ratio r_e of the rectangular region is set within the range of r_1 to r_2. Using S_e and r_e, the height H_e and width W_e of the rectangular region are determined as follows:

$$H_e = \sqrt{S_e \times r_e} \tag{1}$$

$$W_e = \sqrt{\frac{S_e}{r_e}} \tag{2}$$

Finally, the position (x_e, y_e) where the rectangular region is superimposed is randomly determined. The rectangular region I_e is overlaid from (x_e, y_e) to $(x_e + W_e, y_e + H_e)$. The pixel values within the rectangular region are randomly assigned between 0 and 255.

If the overlaid region overlaps with an annotation, the overlapping area S_o with the segmentation mask is calculated. If the overlap ratio, defined as S_o/S_s (where S_s is the segmentation mask area), is below a predetermined threshold, the region remains as is. Otherwise, a new rectangular region is generated. By computing the overlap ratio with the segmentation mask, this method prevents the complete occlusion of parts.

The above process is repeated for a predetermined number of masks n. Fig. 4 shows an example of the mask of the proposed method utilizing random erasing.

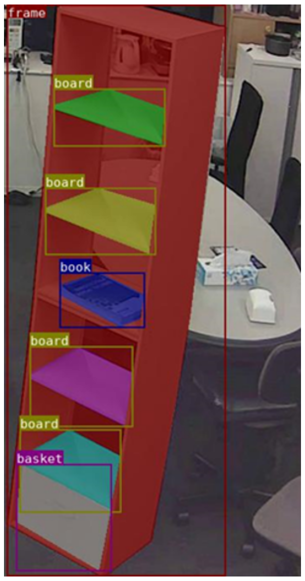


Figure 3. Anotatiom Data

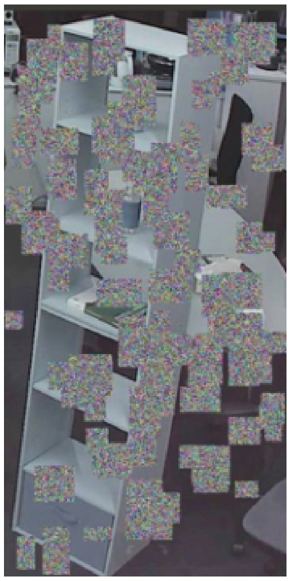


Figure 4. Data Augmentation using the proposed masking

4. EXPERIMENT

4.1 Experiment Environment

Experiments were conducted to estimate the assembly process using the proposed method. Two colored boxes stacked on top of each other were used as the assembly target to simulate the assembly target on site. Video of the assembly process was captured by a web camera. The camera used was a Qwatch TSWR-LP network camera from I-O DATA DEVICE, with a pixel setting of 1980 × 1080. Fig. 5 shows the experimental environment.

4.2 Experimental Conditions

The training data consisted of 332 sheets, the test data consisted of 220 images, the number of training cycles was 200000 iterations, and the batch size was 4. The defined assembly process is shown in Fig. 6. The number of assembly processes was set to 10, from step 0 to 9. The parts to be detected were partition boards, buckets, books, alcohol gels, tumblers, and cloths.

The parameters of the proposed method were set to 100 for the number of rectangular regions, 0.3 for the probability of data expansion, 0.5 to 2.0 for the aspect ratio of rectangular regions, 0.002 to 0.01 for the area of the rectangular region, and 0.1 for the overlap ratio with the segmentation mask. These values were determined empirically. To evaluate the accuracy of the progress estimation on the test data, we compared four methods: YOLACT alone, YOLACT with Random Erasing, Gridmask, and the proposed method. Each was tested five times, and the average accuracy was calculated.

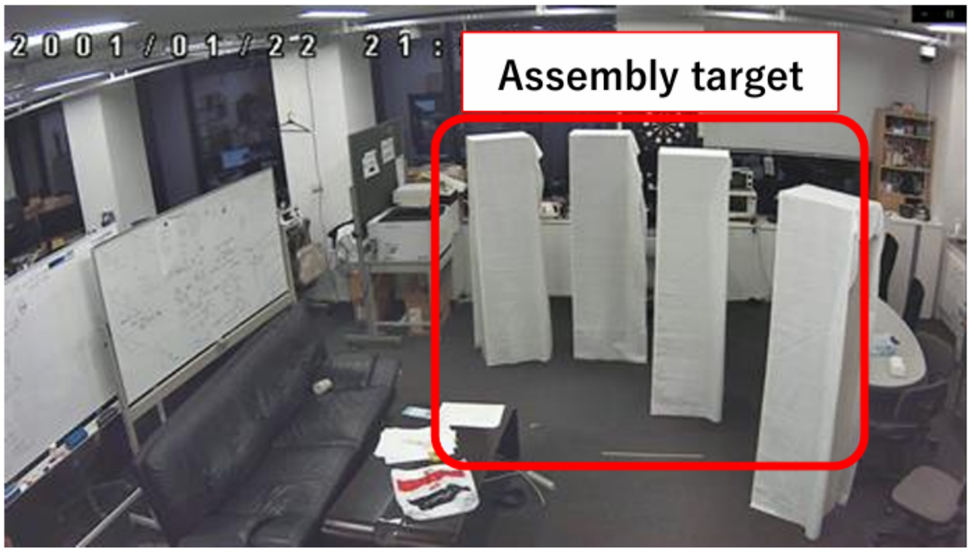


Figure 5. Expeiment Environment

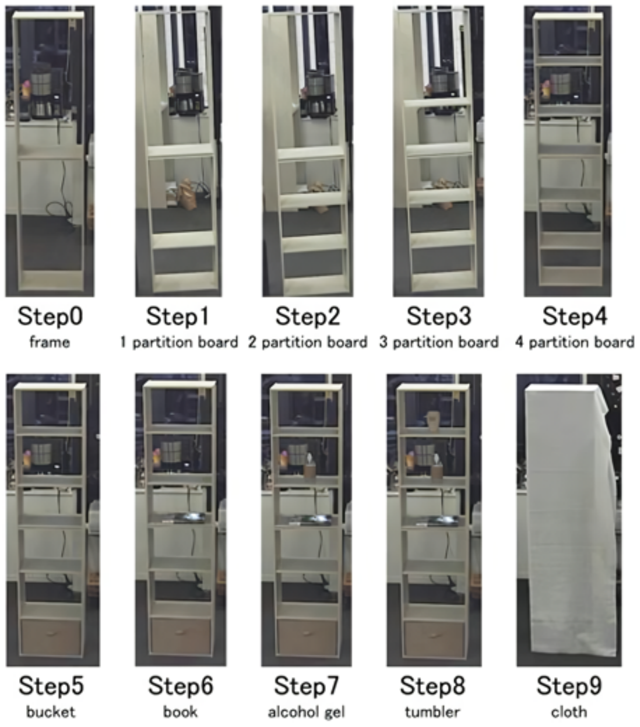


Figure 6. Step of the object assembly

4.3 Experimental Results

First, check to see if the product parts are detected correctly. Figure 7 shows that the parts are detected in the correct position and the mask is generally shown in the correct position. Figure 8 shows incorrect assembly, with the basket installed before the partition board. This incorrect assembly results in a correctly determined "Error".

Table 1 shows the experimental results. The confusion matrices of the YOLACT alone, Random Erasing, Gridmask, and the proposed method are shown in Figs. 9, 10, 11, and 12, respectively. The proposed method achieved the highest accuracy. The model using the existing data extension methods, Random Erasing and Gridmask, and the proposed method showed higher accuracy than YOLACT alone. This suggests that generalization performance is enhanced by data augmentation and the detection accuracy is improved. The accuracy of the proposed method was higher than that of the existing data extension. By comparing the confusion matrix of YOLACT alone and the proposed method, the accuracy of Step 6 is improved by 14 %. Step 6 is the process of installing the book. Compared to other parts, the appearance of a book tends to change depending on the angle of view. Therefore, it is considered that the proposed method reproduces various occlusions by using multiple shields and achieves higher accuracy than existing data expansion methods.

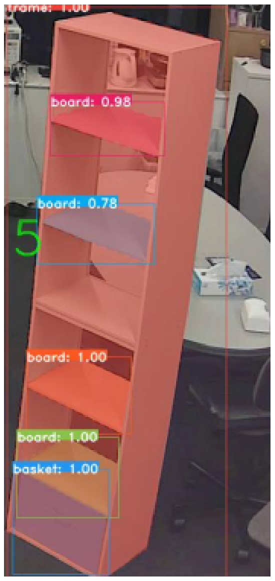


Figure 7. Part detection via semantic segmentation[8]

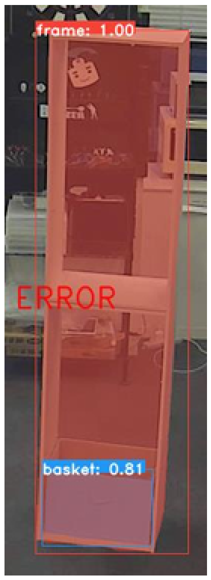
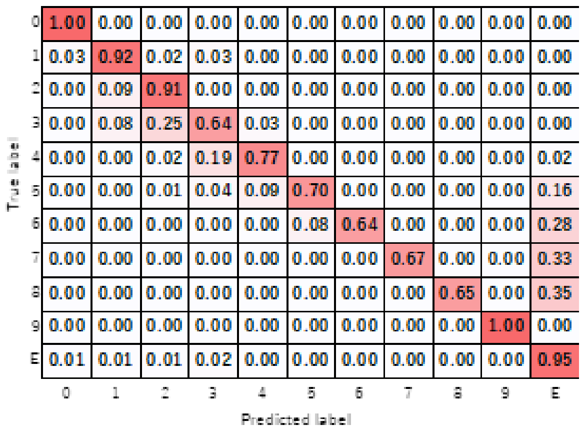


Figure 8. Example of error detection

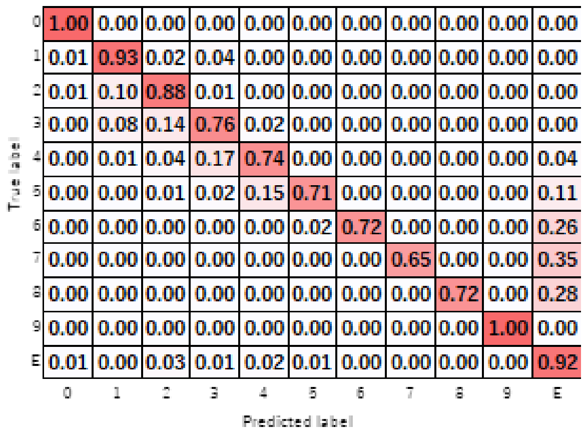
Table 1. Experimental Results

Method	None	Random Erasing	Gridmask	**Proposed Method**
Accuracy	80.5%	82.1%	81.0%	**83.6%**



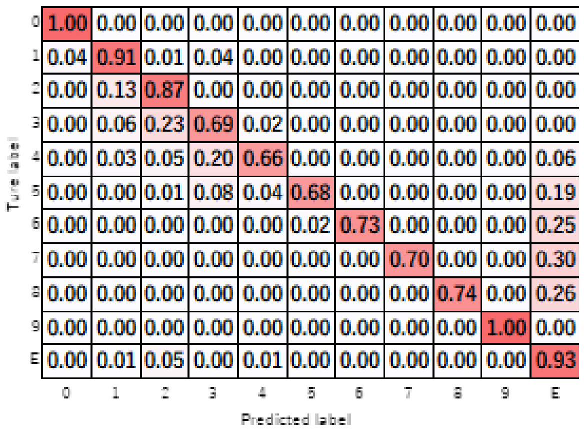
	0	1	2	3	4	5	6	7	8	9	E
0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.03	0.92	0.02	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.09	0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.08	0.25	0.64	0.03	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.02	0.19	0.77	0.00	0.00	0.00	0.00	0.00	0.02
5	0.00	0.00	0.01	0.04	0.09	0.70	0.00	0.00	0.00	0.00	0.16
6	0.00	0.00	0.00	0.00	0.00	0.08	0.64	0.00	0.00	0.00	0.28
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00	0.33
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.65	0.00	0.35
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
E	0.01	0.01	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.95

Figure 9. Confusion matrix of YOLACT alone



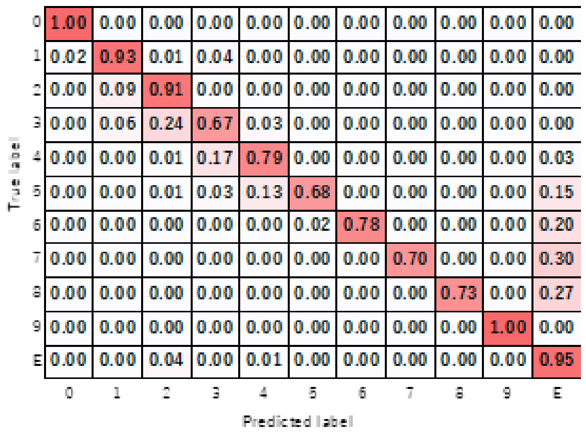
	0	1	2	3	4	5	6	7	8	9	E
0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.01	0.93	0.02	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.01	0.10	0.88	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.08	0.14	0.76	0.02	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.01	0.04	0.17	0.74	0.00	0.00	0.00	0.00	0.00	0.04
5	0.00	0.00	0.01	0.02	0.15	0.71	0.00	0.00	0.00	0.00	0.11
6	0.00	0.00	0.00	0.00	0.00	0.02	0.72	0.00	0.00	0.00	0.26
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.65	0.00	0.00	0.35
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.72	0.00	0.28
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
E	0.01	0.00	0.03	0.01	0.02	0.01	0.00	0.00	0.00	0.00	0.92

Figure 10. Confusion matrix of Random Erasing



	0	1	2	3	4	5	6	7	8	9	E
0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.04	0.91	0.01	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.13	0.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.06	0.23	0.69	0.02	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.03	0.05	0.20	0.66	0.00	0.00	0.00	0.00	0.00	0.06
5	0.00	0.00	0.01	0.08	0.04	0.68	0.00	0.00	0.00	0.00	0.19
6	0.00	0.00	0.00	0.00	0.00	0.02	0.73	0.00	0.00	0.00	0.25
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00	0.30
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.74	0.00	0.26
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
E	0.00	0.01	0.05	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.93

Figure 11. Confusion matrix of Gridmask



	0	1	2	3	4	5	6	7	8	9	E
0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.02	0.93	0.01	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.09	0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.06	0.24	0.67	0.03	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.01	0.17	0.79	0.00	0.00	0.00	0.00	0.00	0.03
5	0.00	0.00	0.01	0.03	0.13	0.68	0.00	0.00	0.00	0.00	0.15
6	0.00	0.00	0.00	0.00	0.00	0.02	0.78	0.00	0.00	0.00	0.20
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00	0.30
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.73	0.00	0.27
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
E	0.00	0.00	0.04	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.95

Figure 12. Confusion matrix of the proposed method

5. CONCLUSION

In this study, we proposed a method for estimating product assembly progress using instance segmentation. The progress is estimated from the type and number of parts detected by YOLACT based on product images. By utilizing data argumentation by random erasing, we proposed a method that enables detection without loss of accuracy even with a small amount of training data. Experimental results confirmed that the proposed method outperforms conventional data expansion methods in terms of accuracy. Prospects for this research include a method that takes advantage of changes in the overlap ratio with the segmentation mask.

For future work, the progress estimation method in this study could be improved to be more general-purpose. The current method estimates which of the pre-defined assembly processes the input image falls into. However, depending on the delivery status of the parts, the assembly order may change, and an undefined combination of parts may be detected and judged as an error. By modifying the definition of errors, the system can become more versatile and handle these situations more effectively.

REFERENCES

[1] Wang J, Ma Y, Zhang L, Gao RX, Wu D, ”Deep learning for smart manufacturing: methods and applications,” Journal of Manufacturing Systems. vol. 48. 2018. pp. 144–156.

[2] M. Funk, A. Bachler, L. Bachler, T. Kosch, T. Heidenreich, ”Working with augmented reality? A long-term analysis of in-situ instructions at the assembly workplace.” Proceedings of the 10th international conference on pervasive technologies related to assistive environments. 2017. p. 222-229.

[3] Hirotomo Oshima, Yuta Shirakawa, Takahiro Yoshii, Takehiro Kado, Takuya Maekawa, and Yasuo Namioka, ”Automatic Extraction of Work Progress Based on Product State Estimation Using Deep Learning,” Proceedings of the Production Systems Division Conference 2022, The Japan Society of Mechanical Engineers, 2022, p. 206. (in Japanese)

[4] T. Kitsukawa et al., ”Camera-based Progress Estimation of Assembly Work Using Deep Metric Learning,” 2023 IEEE/SICE International Symposium on System Integration (SII), Atlanta, GA, USA, 2023, pp. 1-6.

[5] DeVries, T., and G. W. Taylor, ”Improved regularization of convolutional neural networks with cutout,” arXiv preprint arXiv:1708.04552, 2017.

[6] Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y. ”Random erasing data augmentation.” Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 07. 2020. p. 13001-13008.

[7] Chen, P., S. Liu, H. Zhao, and J. Jia, ”Gridmask data augmentation,” arXiv preprint arXiv:2001.04086, 2020.

[8] Bolya, D., Zhou, C., Xiao, F., Lee, Y. J, ”Yolact: Real-time instance segmentation.” Proceedings of the IEEE/CVF international conference on computer vision. 2019. p. 9157-9166.

[9] He, K., Gkioxari, G., Dollár, P., Girshick, R. ”Mask r-cnn.” Proceedings of the IEEE international conference on computer vision. 2017. p. 2961-2969.