Estimation of Object Detection Threshold for Object Tracking by Deep Learning

Kazuma Miura
Cource of Precision Engineering
Chuo University
Tokyo, Japan
miura@sensor.mech.chuo-u.ac.jp

Takumi Kitsukawa
Cource of Precision Engineering
Chuo University
Tokyo, Japan
kitsukawa@sensor.mech.chuo-u.ac.jp

Sarthak Pathak
Dept. of Precision Mechanics
Chuo University
Tokyo, Japan
pathak@mech.chuo-u.ac.jp

Kazunori Umeda
Dept. of Precision Mechanics
Chuo University
Tokyo, Japan
umeda@mech.chuo-u.ac.jp

***Abstract*—This paper proposes an accurate product tracking system in a factory via automated calculation of object detection thresholds. First, the system uses deep learning to estimate the thresholds of parameters used for object detection with images acquired from a fixed-point camera installed in the factory. Next, object detection is performed using the estimated parameter thresholds. Finally, object tracking is performed based on the number of detected objects, assuming that the error of the number is small. The proposed automatic thresholds setting was evaluated experimentally using 67 images captured in a real factory environment. The number of error images decreased from 16 to 12. Additionally, the total count of False Positives and False Negatives reduced from 19 to 13. The experimental results confirm the effectiveness of our proposed thresholds learning method.**

***Keyword*—deep learning, object detection, object tracking, assembly**

I. Introduction

There have been attempts to streamline factory operations, such as scheduling, by capturing the work processes by cameras. This can be achieved by combining object detection and object tracking. Although there are various work processes, in this study, we focus on the work process of assembling a large product that takes several days to complete by humans. In this case, several problems exist. There are general methods for understanding the work process by motion recognition of workers [1]. These methods focus on hand movements and are difficult to use when the work object is large. Moreover, the complexity increases due to individual variations among workers and the large amount of time-series data generated during the assembly process. They makes it difficult to perform motion recognition. Instead of comprehending the work process through the motion recognition of the workers , there are methods to understand the progress of the process from the images of the products [2]. In order to determine the progress of a process from a product image, object detection can be performed on the video image to detect the image area of the product, and the progress of the process can be determined from the image area of the product. In order to focus on the appearance information of the products, accuracy of object detection is required. Faster R-CNN[3] and You Only Look Once(YOLO)[4] are representative object detection methods. In these methods, thresholds are set for the predicted probability at the time of use and for the Intersection over Union(IoU) used when performing Non-Maximum Suppression, in order to improve accuracy. These thresholds are manually adjusted to a constant value based on the results of training and object detection. However, it is difficult to manually set an appropriate threshold for accurate object detection. In addition, because video images are subject to changes in brightness, background, blurring, etc., it is expected that the appropriate thresholds will also change from frame to frame. Therefore, it is important to set appropriate thresholds for object detection for each frame in order to reduce False Positives and to understand the work process from the object image. Object tracking is also valuable for comprehending the product's location and the progress of assembly. Since the accuracy of object tracking depends on the accuracy of object detection, it is essential to achieve accurate object detection. Hence, in this study, we propose a deep learning approach to set an optimal threshold for each frame during object detection. A system for tracking products by setting optimal thresholds for each frame of a video is proposed using deep learning to estimate the optimal thresholds.

II. Related Works

A. Object Detection

In this study, YOLO is used for object detection. YOLO conducts real-time object detection by passing images through a Convolutional Neural Network (CNN), and the detection result is obtained as a Bounding Box (BBox), which is a rectangle enclosing the detected object. The BBox has a Confidence Score as a predictive probability of classifying the detected object. Since the Confidence Score is generally low in areas of False Positives, only those BBox with a

979-8-3503-2099-2/23/$31.00 ©2023 IEEE

Confidence Score above a threshold value are obtained to prevent False Positives and reduce False Positives. To avoid multiple detections on the same object, the overlap between BBox is calculated as IoU, and a threshold is set for IoU. In general, the Confidence Score and IoU thresholds are set uniformly for videos. YOLO is executed based on the established thresholds.

B. Object Tracking

SORT (Simple Online Realtime Tracking)[5] utilizes the Kalman filter and the Hungarian method for object identification and tracking following object detection. However, if the object detection of the previous frame contains a False Positive, the next frame will also incorrectly estimate the object detection. DeepSORT (Simple Online Realtime Tracking with a Deep Association Metric)[6] is an extension of SORT. ByteTrack[7] improves the accuracy of object tracking by considering unreliable detections in the previous frame without discarding them. A common issue with both methods is that they are susceptible to being affected when the results of accurate object detection are not utilized.

III. Proposed Method

Object detection is performed using YOLO. Only when the number of detected objects differs from the previous frame, the number of detections is modified so that it is the same as in the previous frame. This ensures that the number of objects is correctly determined when performing object tracking by using the appropriate thresholds during the YOLO detection phase.

A. YOLO Thresholds Learning with Deep Learning

We use YOLOv5[8] as YOLO. We design a CNN to extract optimal thresholds using each frame. As a dataset, images are extracted from a fixed viewpoint video at regular intervals (1 minute intervals in our experiments). For each image, a different thresholds with a step size of 0.1, ranging from 0.1 to 0.9, is assigned as the Confidence Score and IoU thresholds. A total of 81 threshold combinations (9×9) are used to perform YOLO for detecting two objects: the worker and the target object. Afterwards, the detection results are visually checked and the threshold combinations that correctly detect the objects are selected. If multiple threshold combinations correctly detect objects in a single image, the respective average values of the Confidence Score, IoU, and BBox are calculated. This image and the respective averages of Confidence Score, IoU and BBox are used as the training data set for thresholds optimization. An overview of the constructed network is shown in Fig. 1. The input dataset consists of the image and the average value of the YOLO detection results. After passing through the convolutional layer, these values are input to a fully-connected layer. Finally, the estimated Confidence Score and IoU are output. During training, the BBox information from the prepared dataset is input, and during estimation, the BBox information from running YOLO at the default thresholds are input to obtain the Confidence Score and IoU. As the BBox information input during estimation is considered to contain False Positives and False Negatives, the obtained Confidence Score and IoU are corrected and used for object detection.

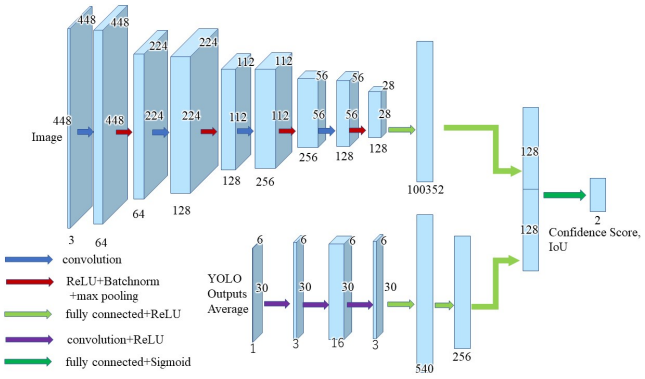

Fig. 1. Network used for thresholds training

B. Object Tracking with Number of Positive Detection

For the first few frames, object detection is performed using the estimated thresholds. The average number of objects detected during that period is calculated. In subsequent frames, if more objects are detected than the criteria based on the calculated number of object detections, the BBox with a low Confidence Score is deleted as False Positives occurred.If the number of detected objects is less than the criterion, the Kalman filter is used to obtain the BBoxes. Compared to the case where the Kalman filter is used without setting a criterion for the number of object detections, this method is expected to prevent using the BBox that were erroneously detected in the previous frame in the next frame. The Hungarian method is applied to the obtained BBoxes to assign IDs. Since the number of detections in the image fluctuates as the operator moves objects, the criteria for the number of detections is adjusted if there are people and objects at the coordinates of the edges of the screen.

IV. Experiment

Experiments were conducted on the learning and estimation of the optimal thresholds for YOLO among the proposed methods.

A. Learning Experiments for Optimal Parameter Estimation

We prepared 388 images from the video of the factory. Due to rights restrictions, we are not unable to share images of the inside of the factory. Instead of the actual images, an illustration of the inside of the factory is shown in Fig. 2. The product being assembled in the factory is a semiconductor inspection equipment. The images have characteristics such as low image quality, some frames with noise, presence of occlusion, similarity in appearance of products, and irregular arrangement of products. 81 thresholds were used for YOLO, and images containing 52 False Positives or False Negatives

were excluded from the data set. The dataset consisted of 336 images that yielded ideal results by varying the threshold combinations. Out of these, 215 images were assigned to train, l54 images to validation, and 67 images to test. For training, epochs = 200, batch size = 4, learning rate = 0.000001. The training was evaluated based on loss and accuracy. Loss was measured using the mean absolute value error. Accuracywas considered correct if the difference between the Confidence Score and the IoU was within 0.025. Fig. 3 and Fig. 4 show the results of training. The training loss consistently decreased, indicating successful learning without overfitting.

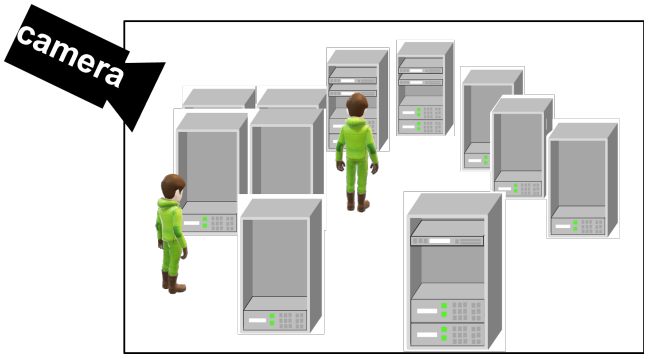

Fig. 2. Illustration of the Inside of the Factory

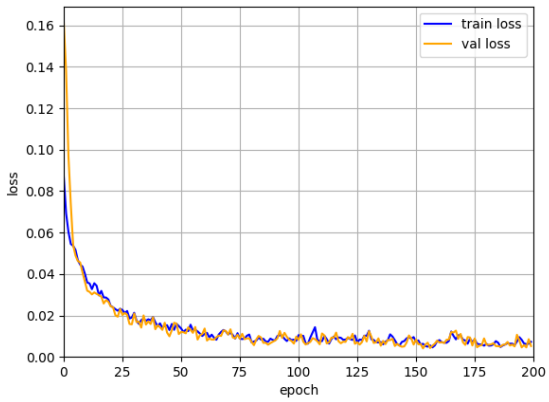

Fig. 3. Evaluation of thresholds training loss

B. *Experiments to evaluate object detection*

We conducted an experiment to evaluate object detection using the estimated optimal parameters. 67 factory images assigned to the test were tested using the proposed trained model. The results are shown in Fig. 5 and Fig. 6. The correlation coefficients between the true values and the estimated values were approximately 0.29 for the Confidence Score and

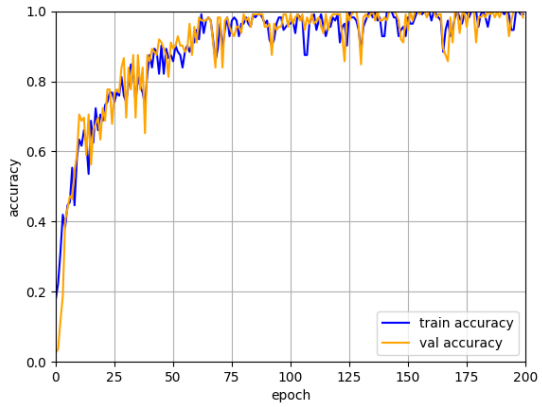

Fig. 4. Evaluation of thresholds training accuracy

approximately 0.068 for the IoU. The reason for the small correlation is considered to be that the true value is determined from the average of the thresholds when there are multiple combinations of thresholds that correctly detect an object.

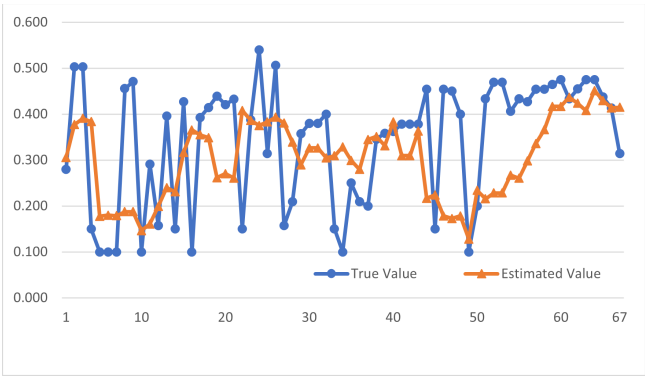

Fig. 5. Evaluation of Confidence Score estimated by the proposed network

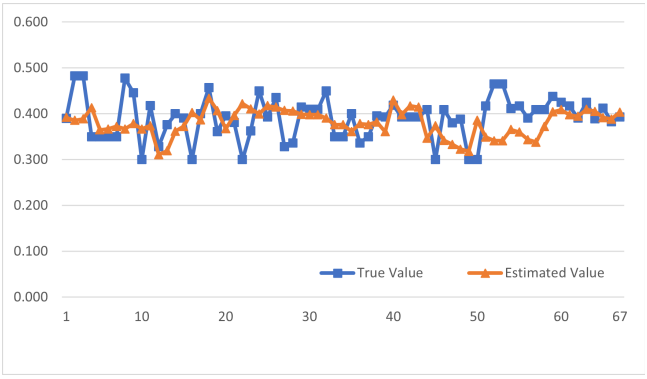

Fig. 6. Evaluation of IoU estimated by the proposed network

Next, object detection was performed using the estimated parameters, and compared to the results obtained with YOLO's default thresholds of Confidence Score=0.25 and IoU=0.45,

TABLE I
RESULTS OF OBJECT DETECTION USING DIFFERENT THRESHOLDS

	Default Thresholds	Estimated Thresholds	Corrected Default Thresholds	Corrected Estimated Thresholds
FP	2	1	12	6
FN	17	16	6	7
FP&FN	19	17	18	13
Error Images	16	14	17	12
Correct Images	51	53	50	55

the number of images with incorrect object detection decreased from 16 to 14, and False Positives and False Negatives were reduced by 1 each, bringing the total from 19 to 17. The results were judged visually. When focusing on the images with incorrect detection results, it was observed that the average difference between the estimated value and the true value of the Confidence Score threshold was about 0.11 larger than the estimated value. Table 1 shows the results of object detection using the estimated parameters. The number of correctly detected images increased compared to the default value before correction, and further increased when the estimated value was corrected. However, comparing the thresholds corrected based on the default value and the thresholds corrected based on the estimated value, it was found that the thresholds corrected based on the estimated value resulted in more images with correct detection. This suggests that the threshold value set for the entire video should not simply be lowered by 0.11 from the default value of Confidence Score, but should be estimated from the image and adjusted accordingly. It was suggested that estimating the threshold value contributes to improving the number of correctly detected frames. Therefore, the proposed per-frame parameter estimation method was found to be effective.

V. CONCLUSION

In this paper, we proposed a system that tracks products by dynamically setting thresholds for each frame of a video using deep learning. Experiments confirmed the system's ability to accurately estimate thresholds for object detection from the images. For object detection on 67 factory images, the number of images with incorrect object detection decreased from 16 to 12, and the total number of False Positives and False Negatives decreased from 19 to 13 against the conventional method. As future work, we plan to implement the object tracking method introduced in Section III-B and evaluate the proposed method as a whole.

VI. ACKNOWLEDGEMENT

We would like to thank Hitachi High-Tech Corporation and Hitachi High- Tech Solutions Corporation for providing us factory data and making this work possible.

REFERENCES

[1] Q. Xia, J. Korpela, Y. Namioka, and T. Maekawa, "Robust unsupervised factory activity recognition with body-worn accelerometer using temporal structure of multiple sensor data motifs", *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, Vol. 4, No. 3, Article 97, pp. 1–30, 2020.

[2] T. Kitsukawa, S. Pathak, A. Moro, Y. Harada, H. Nishikawa, M. Noguchi, A. Hamaya, and K. Umeda, "Camera-based progress estimation of assembly work using deep metric learning", *2023 IEEE/SICE International Symposium on System Integration (SII)*, pp. 1-6, 2023.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks", *Neural Information Processing Systems (NIPS)*, 2015.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B.Upcroft, "Simple online and realtime tracking", *IEEE International Conference on Image Processing (ICIP)*, 2016

[6] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric", *IEEE International Conference on Image Processing (ICIP)*, 2017.

[7] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: multi object tracking by associating every detection box", *Computer Vision – ECCV 2022*, 2021.

[8] G. Jocher, ultralytics/yolov5, http://github.com/ultralytics/yolov5.