Title: Object Recognition Method Using Locus of Gestures Detected by YOLOv5
Author: Tsukasa Kudo

In Proceedings of International Workshop on Informatics (IWIN2022), Aug. 31 – Sep. 3, 2022.

This paper is an extract from the above-mentioned Proceedings, which are available at the following URL.
http://www.infsoc.org/conference/iwin2022/download/IWIN2022-Proceedings.pdf

Object Recognition Method Using Locus of Gestures Detected by YOLOv5

Tsukasa Kudo[†]

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

Abstract - To recognize relatively small objects in an image, it is first necessary to perform object detection. In recent years, research has been actively conducted to utilize deep learning to simultaneously perform object detection and recognition, in which real-time object recognition has been also enabled by such as You Only Look Once (YOLO). However, they are based on deep learning, there are application issues that training data for all targets must be prepared for training the model. In this study, to detect target objects, I propose a method to detect the locus indicated by gestures in videos using YOLOv5, which uses only a single object such as a hand, to extract the target area. In this method, to improve the accuracy of the target area, the false object detection results are eliminated and the locus is corrected, by using the median and moving average of consecutive video frames respectively. Furthermore, it is shown that simple object recognition methods such as template matching can be used by detecting the size and tilt of the target based on the detected area by this method.

Keywords: YOLOv5, Deep learning, Object detection, Gesture recognition, Template matching

1 INTRODUCTION

In recent years, object recognition for videos and images has been actively studied, and its applications are expanding in various fields such as immigration control by face recognition and automatic car driving. However, when the area of the target (hereinafter, target area) in the image is small, it is necessary to perform object detection firstly to specify the target area before object recognition. For example, in the case of face recognition, face detection is performed using Haar-like features, and then face recognition is performed on the detected target area [1].

Using deep learning, various methods have been proposed to efficiently perform both object detection and recognition. For example, You Only Look Once (YOLO) detects the target as a bounding box and simultaneously recognizes the target [2]. Furthermore, it has been shown that even videos can be processed in real-time [3].

However, these methods require the preparation of model training data for each target object, which is a significant burden when the types of target objects are large. On the other hand, for still objects that can be photographed from a specific direction, such as the cover of a book, object recognition can be performed by a simple method such as template matching if the target area can be identified.

In this paper, I propose a method to extract the target area from the gesture in videos by using YOLO for object detection. The gesture is performed so that the target area is a closed area surrounded by the locus of the gesture, and the target area is extracted by using this locus. The important point is that, since gestures can be performed by a certain part of the body such as a hand, only one type of training data is required for a variety of object detections in this method. Also, while other object detection methods extract the target area as a bounding box, this method can extract the target area according to the shape of the target object. That is, for example, when performing template matching, the method can estimate and correct the target's tilt or suppress the influences of background areas.

However, since the above locus is created by continuously detecting a specific part of the gesture in a video, it causes some challenges. The locus contains wrong points due to false detections (hereafter, noises); there may be a double and missing part of the locus at the beginning and end of gestures. To address these challenges, this method eliminates these noises and corrects the locus by utilizing the median and moving average of the locus points. And, I show that this method can extract the target area through experiments.

Furthermore, in order to investigate the effects of size specification and tilt correction on recognition accuracy in template matching, which is one of the simplified object recognition methods, I evaluated the improvement of recognition accuracy for books. The purpose of this evaluation was to clarify the required accuracy in the target area extraction. The results show that when the vicinity of the target region is extracted, with a size error of less than 10% and a tilt error of less than 10°, the detection is correct.

The remainder of this paper is organized as follows. Section 2 presents related works and the aim of this study, and Sec. 3 proposes a target area extraction method based on gestures in a video. Section 4 shows the implementation and experimental results of target area extraction, and Sec. 5 evaluates the accuracy of the proposed method and its effectiveness for template matching. Section 6 discusses on the evaluation results, and Sec. 7 concludes this paper.

2 RELATED WORKS AND AIM OF THIS STUDY

In recent years, the effectiveness of object recognition based on deep learning for images and videos has been widely recognized and applied to various fields. On the other hand, when the target area in an image is relatively small, recognition accuracy deteriorates. So, it is necessary to extract the target area firstly and then perform object recognition.

So, various methods for simultaneously detecting and recognizing objects have been proposed. Faster R-CNN performed both of them in a lump by collective end-to-end training of both models [4], and YOLO executed them with a single neural network to improve efficiency [2]. Concerning different scale objects, SSD made it possible to process them collectively [5], and RetinaNet improved efficiency by introducing the Feature Pyramid Network (FPN) and improving the loss function [6], [7]. Then, M2Det has further improved accuracy and efficiency by introducing the new FPN and loss function [8].

Among these methods, YOLO has been improved repeatedly through version upgrades, and several models are currently available as YOLOv5 [9]. YOLO estimates the bounding box surrounding the target area and the probability of containing the target when the center of the target area is located in a grid cell. The grid cell is a part of the image divided by grids. And, YOLO is known to have high detection efficiency and accuracy. And, it has been shown that YOLO can be applicable to real-time object detection and recognition [3].

However, because the above methods use deep learning, it is necessary to prepare training data consisting of images and correct labels for model training. For example, YOLO requires not only the preparation of images for training the model but also the corresponding correct labels indicating the location and classification of bounding boxes for each object contained in each image. Therefore, when targeting a large number of object types, the burden of creating these labels increases, which is a major obstacle in practical applications.

On the other hand, trained models and training data for YOLOv5 for various objects, such as the coco dataset, are available on the Internet [9]–[11]. Therefore, when targeting specific objects, YOLO can be easily used for the object detection and recognition from videos in real-time.

The motivation for this study is the idea that the target area of an arbitrary object can be extracted, by detecting the locus of a specific object such as the tip of a hand indicated by a gesture in a video. This locus can be detected in real-time by YOLO, targeting only one type of object, that is, training the model is easy. In addition, using the target area extracted in this way, the target size and tilt can also be estimated from the area. In other words, when recognizing still objects viewed from a specific direction, such as back covers of books on a shelf, it is expected that a simpler method such as template matching can be used instead of the methods using deep learning.

Several applications have been proposed for hand gesture recognition using deep learning, such as conversation and device control [12]–[14]. However, I could not find application studies to extract the target area of an object. Furthermore, in continuous object detection using gestures in videos, it is necessary to eliminate noises due to object detection errors and to correct a double or missing part near the beginning and end of the locus.

The aim of this study is to propose a method for extracting the target area with high accuracy using gesture locus and clarify its effects and practical issues for applying it to object recognition.

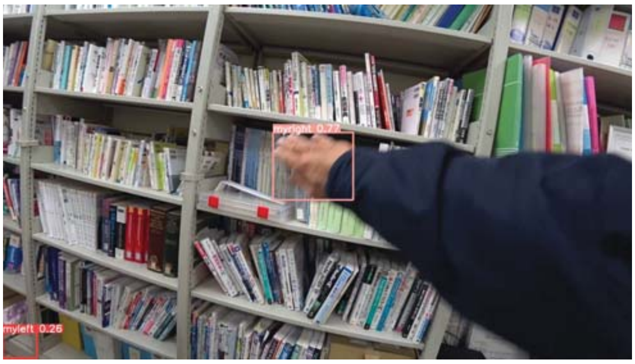


Figure 1: Right hand object detection and recognition using YOLOv5

3 PROPOSAL OF TARGET AREA EXTRACTION METHOD USING GESTURE

3.1 Target Video Frame Images

I propose a method for extracting the target area by using gestures in videos. In this method, the locus of the gesture is detected using YOLOv5 (hereinafter, YOLO) sequentially for each frame of the video, and extracts the area surrounded by the locus for the target area. In order to correctly detect the locus indicated by the gesture, we perform noise elimination and locus correction as mentioned in Sec. 2.

Figure 1 shows an object detection and recognition (hereafter, object detection) result image of the right hand by YOLO from a video frame. The detected target is indicated by a bounding box, and the class of the target and the recognition accuracy are indicated above the upper side of the box. In this image, my right hand (“myright”) is detected with an accuracy of 77% (“0.77”) in the center. On the other hand, the lower-left bounding box is falsely detected noise, and its accuracy is 26%. Therefore, in this case, the former is adopted.

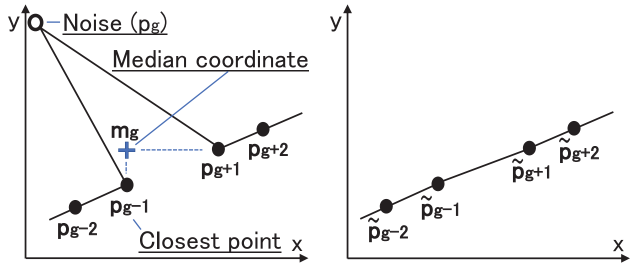
3.2 Target Area Extraction Procedure

The procedure to extract the target area from such as the image of Fig. 1 is shown as follows. Valid coordinates of the locus are selected, and noises are eliminated using the coordinates median of adjacent frames. Then, the locus is corrected using the moving average.

3.2.1 Selection of Valid Coordinates in Locus

In this method, one of the vertices of the bounding box shown in Fig. 1 is selected as a point constructing the locus of the gesture. In the following, it is assumed that the coordinates of the top-left vertex are selected. The valid bounding box is selected in each image with the following condition: its accuracy is the highest in the image and greater than the threshold namely the specified value. And, its coordinate of the top-left vertex is adopted for the locus as the valid coordinate.

If each set of the coordinate and accuracy of the i-th frame is indicated by c_{ij} and a_{ij}, the valid coordinates s_i shown in



(1) False trajectory (2) Corrected trajectory

Figure 2: Noise reduction by median coordinate

the Eq. (1) is selected.

$$s_i = \begin{cases} c_{ik} & (\exists k(a_{ik} \geq L \wedge a_{ik} = max(\{a_{ij}\})) \\ \emptyset & (\forall k(a_{ik} < L)) \end{cases} \quad (1)$$

Here, $\emptyset$ indicates that the coordinates of the frame are not selected; L indicates the threshold; $\{a_{ij}\}$ indicates the set of a_{ij}. In the case of Fig. 1, if $L = 0.4$, then the upper-left coordinate of the center bounding box is selected because its accuracy is the highest and greater than the threshold.

3.2.2 Noise Elimination Using Median Coordinates

To eliminate noises, the median coordinates are calculated from the valid coordinates of the previous and next frames. Let $p_g(g = 1, 2, 3, \cdots)$ be the ordered set of coordinates with eliminating s_i if $s_i = \emptyset$ from $\{s_i\}$ the set of s_i in Eq. (1). Figure 2 (1) shows an example where p_g is a noise.

The median of p_g is constructed using the interval before and after the index g. Let R_g indicates the set of indices of this interval, and let p_{Rx} and p_{Ry} indicate the set of x-coordinates and y-coordinates, respectively. I define the median coordinate of this interval by $m_g = (median(p_{Rx}), median(p_{Rx}))$. Here, *median* is the function to get the median value of the coordinates. And, in the case of Fig. 2, the coordinate “mg” with the median of each of the x-y coordinates is selected.

The noises are eliminated by using these median coordinates. As shown in Eq. (2), if p_g is not the closest coordinate to m_g for the interval R_g, then it is converted to a coordinate $\tilde{p}_g$ with empty $\emptyset$; else p_g is adopted for $\tilde{p}_g$.

$$\tilde{p}_g = \begin{cases} p_g & (\tilde{m}_g = p_g) \\ \emptyset & (\tilde{m}_g \neq p_g) \end{cases} \quad (2)$$

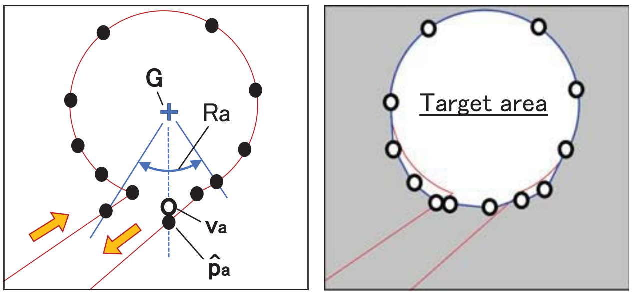
Here,

$$\tilde{m}_g = \{p_h | \exists h(dist(p_h, m_g) = min(dist(p_n, m_g)) \\ \wedge \forall n \in R_g)\}$$

$dist(p_n, m_g)$ indicates the distance between p_n and m_g. In other words, $\tilde{m}_g$ denotes the coordinate of $p_n(r \in R_g)$ that is closest to the median coordinate m_g; and, if the corresponding point p_g is not this coordinate, then it is set to $\emptyset$. The coordinates of the locus without noises are obtained by eliminating $\emptyset$ from the set of coordinates $\{\tilde{p}_g\}$.

In the case of Fig. 2, the closest coordinate to the median $\tilde{m}_g$ is p_{g-1}, so the coordinate $\tilde{p}_g$ is set to $\emptyset$ and eliminated. The other coordinates of are set as $\tilde{p}_n = p_n(r \in R_g)$. As a result, a locus without the noise is constructed as shown in Fig. 2 (2).

Remarks) ✚: Center of gravity (G); ○ : Moving average point (va); ● : Points on locus with noise excluded (p̂a).



(1) Whole locus (2) Target area extraction

Figure 3: Target area extraction using moving average

3.2.3 Target Area Extraction by Using Moving Average

To compensate for doubles or missing near the beginning and end of the locus, a moving average of the coordinates along the angle from the center of gravity is created. First, the x-y coordinate of the center of gravity G is obtained as a simple average of the x-coordinate and y-coordinate of the set of coordinates $\{\tilde{p}_g\}$ excluding noises ($\emptyset$), respectively.

Next, as shown in Fig. 3 (1), the coordinates of each point are transformed into a pair $\hat{p}_a = (\theta_a, r_a)$ of angle and distance from G. Let R_a be the interval to calculate the moving average corresponding to $\hat{p}_a$ and define the moving average v_a by Eq. (3).

$$v_a = (\theta_a, \bar{r}_a) \quad (3)$$

Here,

$$\bar{r}_a = (\sum r_u)/n \qquad (u \in R_a)$$

The n is the number of coordinates contained in R_a, and it is 5 in the case of Fig. 3 (1). In other words, $\bar{r}_a$ is the average of the distances between G and the coordinates $\hat{p}_a(a \in R_a)$.

By connecting the coordinates of this moving average set $\{v_a\}$ along the angle, the target area is extracted as shown in Fig. 3 (2).

4 IMPLEMENTATION AND EXPERIMENTS

4.1 Implementation

The experimental system was constructed to verify that the proposed method can extract the target area. This system was implemented on a Windows 10 PC, Python Ver. 3.8.13 as the program, and Pytorch Ver. 1.7.1 with CUDA Ver. 11.5 to use YOLO, OpenCV-Python Ver. 4.5.5.64 for image and video manipulation.

YOLOv5s, a highly efficient model of YOLO, was used and was implemented by adding the necessary functions to the publicly available program [9]. Similarly, the publicly available “Egohand Dataset” [11] was used for the training data of the model. This is the data for training the model to

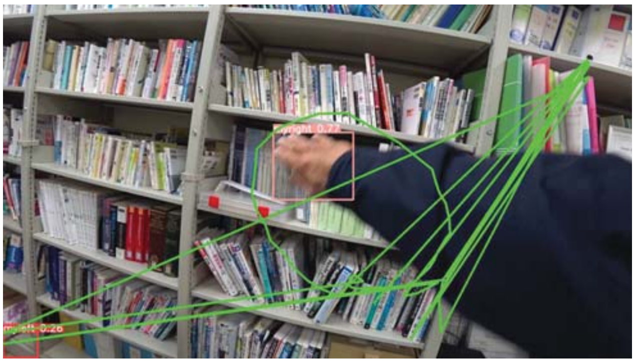


Figure 4: Gesture locus and target area extraction experiment using proposed method

detect four types of hands, the left and right hands of oneself and the other party, and the number of data is 3,840.

Using a model trained with this data, I implemented a program to extract the target area from a video of hand gestures. First, the hands are detected at each frame, and the valid coordinate in locus is selected using the procedure shown in Sec. 3.2.1. In this implementation, the right hand was used, and the upper-left corner was assumed to be the tip of the hand, as shown in Fig. 1.

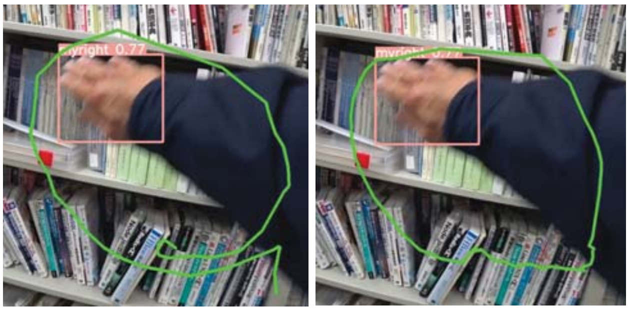
Next, the procedure mentioned in Sec. 3.2.2 and 3.2.3 is used to eliminate noises by median coordinate and extract the target area by moving average. Five points were used to calculate each median coordinate, including the target point and its front and rear points. Since there was no point on one side of the endpoints, their median coordinates are omitted. For the next point, the median coordinate was calculated with three points instead of five points. The moving average is also calculated using 5 points, and the set of moving averages $\{v_a\}$ is obtained. The target area is extracted from $\{v_a\}$ using OpenCV's fillConvexPoly function.

Finally, the bounding box containing the target area is extracted, setting the target area to the frame image and the outside to white.

4.2 Experiments

Using the implemented program, I conducted an experiment to extract the locus of the tip of my hand (hereinafter, hand) captured on video. To confirm that the program can detect even in the case of complex backgrounds, I used the bookshelf shown in Fig. 1. I used a SONY FDRX3000 action camera, and shot videos at $1,920 \times 1,080$ pixels and 30 fps. The hands were moved in a clockwise circular motion starting from the lower right in the image. The accuracy threshold L was set to 0.4.

Figure 4 shows the locus constructed by selecting the coordinates with the highest accuracy for each frame. That is, this is the original locus detected by YOLO. For the background in Fig. 4, the image of Fig. 1 is used. In addition, in the case of this frame image, the hand detected in the center was adopted, so the left-top vertex of its bounding box is on the locus. As shown in Fig. 4, since there were noises due to false detections, the target area could not be extracted directly from this locus.



(1) Locus of median (2) Locus of moving average

Figure 5: Original gesture locus detected by YOLO

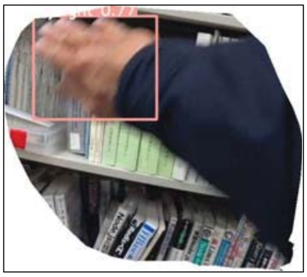


Figure 6: Target area extracted by proposed method

Figure 5 (1) shows the locus after selecting only the valid coordinates and eliminating noises by using median as described in Sec.3.2.1 and 3.2.2, respectively. Note that only the vicinity of the locus has been extracted from the whole image. The noises were eliminated, but the locus was doubled near its beginning and end. Figure 5 (2) shows corrected locus from the one shown in Fig. 5 (1) by using the moving average mentioned in Sec. 3.2.3. The moving average corrected the locus to the place between the doubled loci, and a closed area could be constructed. However, near the endpoints, since the hand locus deviated from the target area, an extra area was included as shown right-lower part.

Figure 6 shows the bounding box of the target area extracted using the locus of Fig. 5 (2). The outside of the target area has been transformed to white.

We performed the above procedure three times in the same environment to examine the number of frames detected at each stage of the procedure. Figure 7 shows the results, and

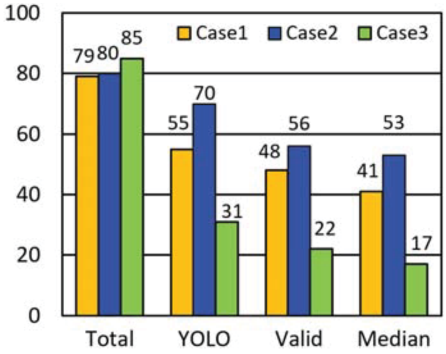


Figure 7: Corrected locus by proposed method

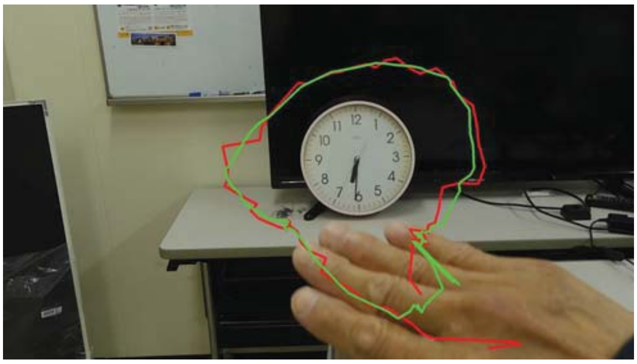


Figure 8: Evaluation of target area extraction accuracy

Figs. 4 to 6 correspond to “Case2”. The vertical axis indicates the number of the detected frames, and the horizontal axis indicates the stage. “Total” shows the total number of frames, “YOLO” shows the number of detected by YOLO including noises, and “Valid” shows the number of accuracies above the threshold (0.4). “Median” shows the number obtained by using medians, that is, the number of frames after eliminating noises, and the number after the moving average is also the same. Note that since the two points at both endpoints of the locus are excluded in the Median stage, as mentioned in Sec.4.1, two points are also excluded for the numbers in the other stages.

As shown in Fig. 7, there was a large difference in the proportion of detections even under similar conditions. In Case3, the proportion detected by YOLO was less than half that of Case2; conversely, Case2 had the highest number of points eliminated due to accuracy under the threshold at the Valid stage. The number of points judged as noise in the Median stage was 7 in Case1 while it was 3 in Case2. The former’s percentage of the total (48), was 14.6%.

5 EVALUATION OF TARGET AREA EXTRACTION AND OBJECT RECOGNITION ACCURACY

5.1 Evaluation of target area extraction accuracy

To evaluate the accuracy of extracting the target area when using hand gestures, I evaluated the extraction accuracy using a round wall clock. In this experiment, the camera was fixed and the hand was moved while watching the monitor in order to evaluate the accuracy of the locus indicated by hand. The used camera was a Nikon COOLPIX A1000, and the resolution and frame rate were the same as in the experiment of Sec. 4.2.

In Fig. 8, the red line shows the locus constructed by using the median; the green line shows the one by using the moving average. The accuracy of the target area is low with respect to the target clock, and in this case, it is outside. Furthermore, at the lower-right part namely near the endpoints of the gesture, it is quite outwardly displaced. The former was caused by using the hand for the gesture, which was too large compared to the target object. The latter was caused by the detection of extra hand movement near the beginning and end of the gesture, namely the movement between the target and the external of the image.

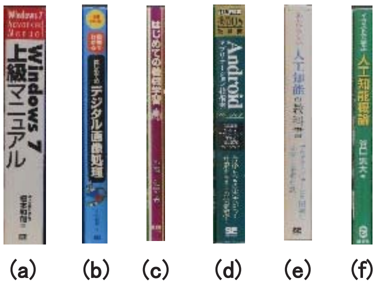
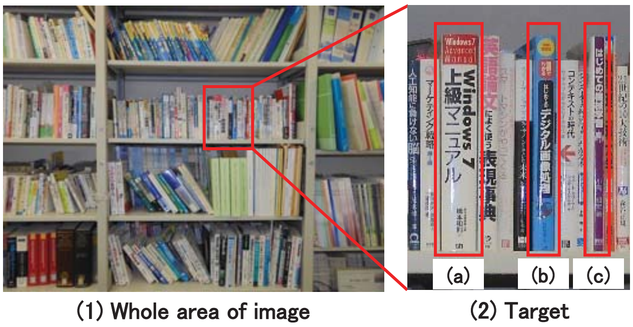


Figure 9: Objects used for template matching



(1) Whole area of image

(2) Target

Figure 10: Images for evaluations of size errors

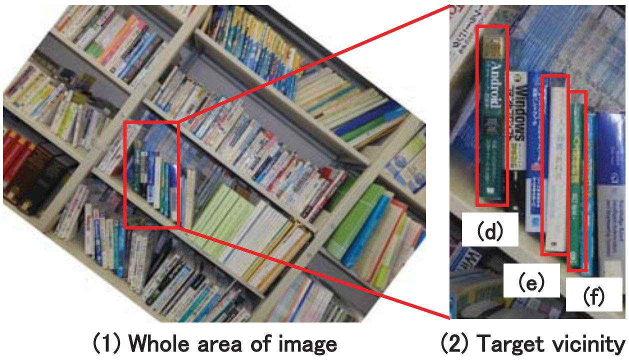
5.2 Evaluation of Object Recognition Accuracy Improvement

To investigate the impact of target area extraction on object recognition accuracy, I evaluated the recognition accuracy of books stored on bookshelves using template matching. The purpose of these evaluations is to clarify the accuracy required to extract the target area by the gesture.

For template matching, the “matchTemplate” function of OpenCV was used with the normalized squared difference matching method. The books to be recognized are the six books shown in Fig. 9. For these books, we evaluated the variation in accuracy when there are errors in size and tilt between the template and the target objects in the image, and when the range of the images was narrowed to the vicinity of the target.

Figure 10 shows the images to evaluate the case of size error. Figure 10 (1) shows the whole image; Fig. 10 (2) shows the image with the narrowed area. Though the latter size is enlarged in this figure, both images are the same size in this experiment. The numbers (a), (b), and (c) below each object correspond to Fig. 9. Similarly, Fig. 11 shows an image to evaluate the case of tilt error, and note that the margins created by the rotation are filled in with white. In addition, the images in Figs. 10 and 11 were shot in a different environment from that of the template image in Fig. 9.

Figure 12 shows the template matching results of all the images in Fig. 9 against Figs. 10 (1) and (2), namely the case of size errors. The “Whole” and “Vicinity” in Fig. 12 corre-



(1) Whole area of image (2) Target vicinity

Figure 11: Images for evaluations of tilt errors

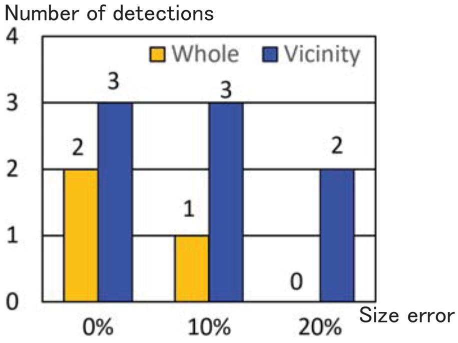


Figure 12: Improvement results for size errors

spond to (1) and (2) in Fig. 10, respectively. “0%” indicates the case where the template size is adjusted to the target of Fig 10, while “10%” and “20%” indicate the case where the template is enlarged to this size, respectively. As shown in Fig. 12, the recognition accuracy degraded as the size error increased, and in the case of the “Whole”, no image was recognized at “20%”. On the other hand, the recognition accuracy in the case of “Vicinity” was improved, and two images were recognized even at “20%”.

Similarly, Fig. 13 shows the result of evaluating the tilt errors, in which “Tilt error” corresponds to the magnitude of the error between the objects in Fig. 9 and Fig. 11. “0°” indicates the case where the image is rotated so that the books are vertical, while “5°” and “10°” indicate the case where the rotation is insufficient by this angle, respectively. As shown in Fig. 13, the recognition accuracy degraded as the tilt error increased, and in the case of the “Whole”, no image was recognized at “10°”. On the other hand, similar to Fig. 12, the recognition accuracy of the “Vicinity” was improved, and two images were recognized even at “10°”.

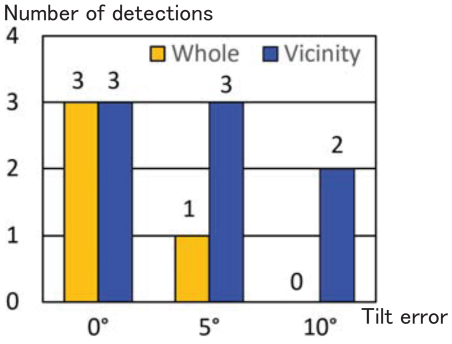


Figure 13: Improvement results for tilt errors

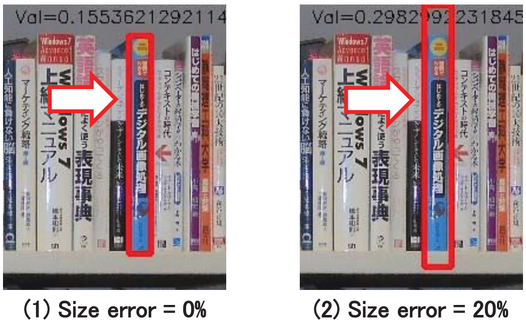


Figure 14: Template matching results on size error

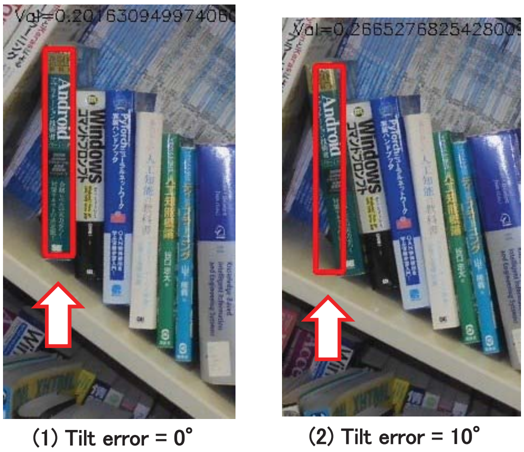


Figure 15: Template matching results on size error

In addition, in the case shown in Fig. 12, the books (d) to (f) shown in Fig. 11 (2) are not detected; conversely, in the case shown in Fig. 13, the books (a) to (c) shown in Fig. 10 (2) were not detected. This is because the tilt of books was too large against the corresponding template shown in Fig. 9, respectively.

Figures 14 and 15 show examples of the matched images for (2) of Figs. 10 and 11, where the matched places are indicated by the rectangles. As shown in (1) of Figs. 14 and 15, when the errors of scale and tilt are small, the extractions were accurate. However, when these errors were large, the accuracies were degraded due to the inclusion of areas other than the target, as shown in (2) of Figs. 14 and 15.

6 DISCUSSION

In this study, I am trying to clarify the effectiveness of the proposed target area extraction method and the issues for its practical applications.

As shown in Fig. 4, several noises tend to be included in the locus due to false object detections used for the gesture. As shown in Fig. 5 (2) and Fig. 8, the proposed method was able to extract the target area by eliminating these noises. Therefore, I consider that the proposed method is effective for the purpose of target area extraction.

As mentioned in Sec. 2, object recognition accuracy can be improved by specifying the target area in the image, namely by object detection. In addition to this, this method can extract arbitrarily shaped target areas, so it is possible to detect the target size and tilt by utilizing the target area. As shown in Figs. 12 and 13, it was possible to improve the recognition accuracy even in simple object recognition such as template matching by using this data,

On the other hand, through experiments and evaluations, the issues for practical use were found, too. The first issue is object detection accuracy including the position of the object used in the gesture. By extracting the target area using gesture, for example, the "vicinity" shown in Figs. 14 and 15 can be applied for the template matching case. In this case, with a size error of less than 10% and a tilt error of less than 10°, the detection was correct. However, as shown in Fig. 8, the gesture locus deviated from the target area. In addition, as shown in Fig. 7, the object detection accuracy differed greatly for even similar gestures.

The former was due to the hand being too large to specify the target area. The latter was due to the use of existing training data. In other words, it is considered that there were some differences in visibility between the images of existing training data and the images of the gesture, though they are the same objects namely hands.

To address these problems, the following measures can be considered. First, we can use the part of the body that can specify the target area more precisely, such as the fingertip, for the gestures. Second, for efficiently model training, we can use transfer learning based on the model trained with the existing training data used in this study according to the usage part of his/her body. In addition, for distant objects, it is considered effective to perform gestures while monitoring with a wearable camera to suppress the difference in viewpoint between the camera and the operator.

The second issue is that, as shown in Fig. 5 (1), the extra locus is detected. This method aims to automatically extract the target area from the continuously shot videos. However, I found that extra hand motions before and after the target gesture are also detected as a part of the gesture. To address this issue, for example, it is considered to stop the gesture at the beginning and end of the target gesture and identify the extra frames in videos.

The construction and evaluations of these measures are the subjects of the next study.

7 CONCLUSION

In order to recognize a small object in an image, it is first necessary to detect the object, and various methods for simultaneous object detection and recognition such as YOLO have been proposed. However, since these methods utilize deep learning, there is an application problem that it is necessary to prepare the training data for each target object.

For this problem, I propose a method to detect the locus indicated by hand gestures in videos using YOLOv5 and extract the target area. Experiments have shown that this method can eliminate noises due to false detection and improve the accuracy of target area detection. Furthermore, I evaluated the effectiveness of this method for template matching and showed that the recognition accuracy can be improved by detecting the target size and tilt in addition to the target area.

However, it was found that further improvement in the target area detection was necessary to improve this recognition accuracy. So, future studies include improving the accuracy of the target area to be extracted by gestures.

REFERENCES

[1] 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, I–I (2001).

[2] 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).

[3] Z. Wang, L. Jin, S. Wang, and H. Xu, "Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system," Postharvest Biology and Technology, Vol. 185, No. 111808 (2022).

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems, pp. 91–99 (2015).

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," European Conf. Computer Vision, pp. 21–37, Springer. (2016).

[6] 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).

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

[8] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," Proc. AAAI Conf. Artificial Intelligence, Vol. 33, pp. 9259–9266.(2019).

[9] G. Jpcher, et.al. "YOLOv5," https://github.com/ultralytics/yolov5 (referred May 17, 2022).

[10] T. Y. Lin, et.al., "Microsoft coco: Common objects in context," European Conf. Computer Vision, pp. 740–755 (2014).

[11] S. Bambach, S. Lee, D. Crandall, and C. Yu, "Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions," IEEE Int. Conf. Computer Vision (ICCV), pp. 1949-1957 (2015), https://public.roboflow.com/object-detection/hands (referred May 19, 2022).

[12] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: a review of techniques," J. Imaging, Vol. 6, No. 8, 73 (2020).

[13] A. Mujahid, M. J. Awan, A. Yasin, M. A. Mohammed,

R. Damaševičius, R. Maskeliūnas, and K. H. Abdulkareem, “Real-time hand gesture recognition based on deep learning YOLOv3 model.” Applied Sciences, Vol. 11, No. 9, 4164 (2021).

[14] Y. Shi, Y. Li, X. Fu, K. Miao, and Q. Miao, “Review of dynamic gesture recognition,” Virtual Reality & Intelligent Hardwar, Vol. 3, No. 3, pp. 183–206 (2021).