In/Out Judgement by Ball Tracking in Table Tennis

Yuta Fujihara[1,*], Xiangbo Kong[2,†], Ami Tanaka[1,†], Hiroki Nishikawa[3,†] and Hiroyuki Tomiyama[1,†]

[1]*Graduate School of Science and Engineering, Ritsumeikan University*
[2]*Department of Intelligent Robotics, Factory of Engineering, Toyama Prefectural University*
[3]*Graduate School of Information Science and Technology, Osaka University*

Abstract
In this study, we focus on the edge ball, and propose a vision-based method for determining whether a table tennis ball bounces on the table including edge balls, or goes out of bounds. For the judgement, we calculate the trajectory based on the ball detection by deep learning, and examine two methods based on the information from them. The first is to express the trajectory by two linear functions and judge from the angle formed by them. The second is to express the trajectory by one quadratic function in addition to the two linear functions and judge from the mean square error calculated by the detection point of the ball and the trajectory. As a result of evaluation, the former sometimes misjudged the edge ball. On the other hand, the latter judged correctly the edge ball which misjudged the former. Therefore, in the proposal, it is better to use the error of the detection point and trajectory of the ball to determine whether batted balls bounce on the table including edge balls, or go out of bounds. In addition, the position of the camera that takes videos must set so as not to disturb the match. Therefore, we change the distance between the camera and the table tennis table to 1.0m, 1.5m, and 1.0m for taking videos and judging. As a result of experiments, the judgement was sometimes incorrect due to an increase in error. However, the accuracy of the judgement was increased by limiting the range of ball detection.

Keywords
table tennis, automatic umpire, Mask R-CNN, trajectory approximation

1. Introduction

In recent years, with the development of AI and machine learning, computer vision technologies such as object detection are applied in various fields. One of the fields is sports. Since it is possible to judge things more accurately than the human eyes, many sports introduce and examine systems to assist referees or automatic referees. Table tennis is one of them. In the T League, a Japanese league, a video judgement system developed by the Sony Group was operated on a trial basis in some matches during the 2019-2020 season [1]. However, the system is difficult to use in amateur matches because it requires expensive cameras. Therefore, we aim to develop a small-scale automatic referee system for amateur matches. In table tennis,

The 5th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2023), August 28-29, 2023, Kusatsu, Japan
*Corresponding author.
†These authors contributed equally.
ri0105kk@ed.ritsumei.ac.jp (Y. Fujihara); kong@pu-toyama.ac.jp (X. Kong); a-tanaka@fc.ritsumei.ac.jp (A. Tanaka); nishikawa.hiroki@ist.osaka-u.ac.jp (H. Nishikawa); ht@fc.ritsumei.ac.jp (H. Tomiyama)

there is an edge ball in which the batted ball hits the corner of the table, and depending on the trajectory of the ball, it may be difficult for human eyes to judge. It causes misjudgements even in official matches [2]. However, while there are some studies related to assisting umpires and making automatic judgements in table tennis, none of them have focused on edge balls as far as I can find. In this study, we focused on this edge ball and propose a vision-based method for determining whether a table tennis ball bounces on the table including edge balls, or goes out of bounds. For the sake of expression convenience, we define the former as "in" and the latter as "out." Because there is no difference in the rule between the ball hits the top and edge of the table, the judgement is made based on two categories: whether the ball is "in" or "out."

In the early stage of the experiment, the position of the camera was taken at a distance of 0.1 m from the table tennis table in order to make easy to detect the ball and process. However, in actual matches, it is necessary to take videos from a little distance away so as not to disturb the playing. Therefore, as a next stage experiment, we consider the processing and decision accuracy when the distance between the camera and the table tennis table is separated.

The structure of this paper is as follows. Section 2 shows the related research of this study. Section 3 describes the method to make "in/out" judgement. Section 4 describes the shooting and processing in which the distance between the table tennis table and the camera is changed. Finally, Section 5 summarizes this paper and discusses future issues.

2. Related Work

There are many studies using deep learning to detect objects including table tennis balls. These are the typical examples of object detection [3][4]. In addition, due to the improvement of computer vision technology using AI, research on images and videos of table tennis is also conducted as a research for practical application [5][6][7].

One of the active research on images and videos of table tennis is ball detection and tracking. In the study [8], two cameras are used to take videos on a table tennis table, and ball detection is carried out based on the difference images obtained from them. In addition, trajectory prediction is carried out from the detected ball by applying a mechanical equation. Further development of this research can be found in the study [9]. In the study [9], in addition to the prediction of the trajectory of the ball proposed in the study [8], the system is developed to track the position of the racket when the ball is hit and to estimate the hitting point. On the other hand, in the study [10], the detection of the ball and the prediction of the trajectory are estimated by a neural network. For the detection of the ball, CNN (Convolutional Neural Network), which is often used in object detection as represented in the studies [3][4], is used. And for the prediction of the trajectory, RNN (Recurrent Neural Network), which can consider time series information, is used. In this way, multiple approaches are studied for ball tracking.

There are also several studies on automatic refereeing and assisting referees. A system for assisting in measuring the height of the toss when serving are devised in the study [11]. In table tennis, there is a rule that the toss of the serve must be raised by 16 cm, and this system is positioned as a system for making this decision automatically. In the study [12], a system for acquiring various information in the video of table tennis in real time is created. Specifically, the system performs multiple tasks in real time, such as ball detection, image segmentation of

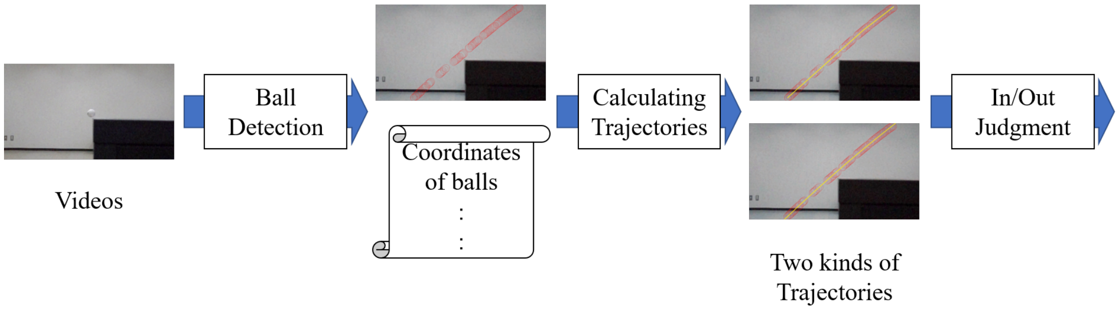

Figure 1: Flow from the Video of the Batted Ball to the In/Out Judgement

table tennis tables and players, and detection of net contact and ball bounce. However, although this study broadly recognizes events that occur during table tennis matches, it is limited to providing information to assist referees.

In this way, there are many studies on table tennis, but few on automatic refereeing, and no studies on ”in/out” decisions for batted balls, including edge balls, existed as far as I investigate. Therefore, in this paper, we focus on edge balls in the video analysis of table tennis, and propose a system to automatically determine ”in/out” of batted balls.

3. In/Out Judgement for Batted Ball

In this section, we describe a method to make ”in/out” judgement for batted balls proposed in this paper.

3.1. Overview

As mentioned above, the process up to the ”in/out” judgement for batted ball is largely divided into three parts. The outline figure is shown in Figure 1. In the proposed method, the batted ball is detecting by using deep learning from the video. Then, the trajectory is calculated by two methods from the detection point of the ball, and the ”in/out” of the batted ball is judged from the trajectory. Details of each process are described below.

3.2. Ball Detection

3.2.1. Instance Segmentation

There are various methods for detecting objects, we decide to detect the ball by using segmentation. There are several types of segmentation, but this time, we do not need to know where something is for the whole image, but we want to detect only the ball. Therefore we use instance segmentation, which performs segmentation only for the specific object we want to detect.

In order to use deep learning, it is common to prepare a large number of images for learning, and it is necessary to prepare images and learning time. Therefore, we use a learned model that has already been learned. There are several ways to use learned models, but we use the one provided by PyTorch, a library for deep learning [13]. We use a type of instance segmentation called Mask R-CNN [14]. The model used was learned on a dataset called COCO [15].

3.2.2. Reshaping Segmentation Results

In this way, ball detection can be easily done using a model of learned instance segmentation. However, in object segmentation, a value is not calculated like a perfect circle. Therefore, the segmentation result is changed into a circle in order to calculate the center coordinates of the circle required for trajectory calculation.

Since the center of gravity of the circle coincides with the center, the center is easily determined by the center of gravity of the area. The segmentation result is expressed by a grayscale image showing the probability of the existence of objects. When this probability is regarded as density, the center of gravity for the area of the ball is calculated in the grayscale image. It is the center when the area is considered as a circle is obtained.

The moment of the image is used for calculating. The moment of the image $M_{i,j}$ is calculated as shown by the following Equation (1) by using the moments function provided by OpenCV [16].

$$M_{i,j} = \sum_{x} \sum_{y} x^i y^j I(x, y) \tag{1}$$

x and y are the coordinates in the image and $I(x, y)$ is the pixel value of the specified coordinates. The calculation of the sigma means the sum of these values in the region of interest. From this equation, the center of gravity (C_x, C_y) is calculated from $M_{0,0}, M_{0,1}, M_{1,0}$ by the following Equation (2).

$$\left(C_x, C_y\right) = \left(\frac{M_{1,0}}{M_{0,0}}, \frac{M_{0,1}}{M_{0,0}}\right) \tag{2}$$

It is also necessary to calculate the radius in order to form a circle, although it is not related to the calculation of the trajectory. Although various calculation methods can be considered, it is not necessary for subsequent calculations, so it is simply determined from the size of the bounding box surrounding the detected ball with a square. Ideally, the box should be square because the detection target is circular, but because it is actually rectangular, we regard the average of each length of x and y of the box as the diameter, and then we divide it by two to get the radius.

3.3. Trajectory Calculation

As described above, we calculated the center of the ball from the moving image of the batted ball, and then we calculate the trajectory by using the center point. As for the expression method of the trajectory, we consider two linear functions and one quadratic function.

For the two linear functions, we divide the trajectory at a certain point and approximating each with a different linear function. The specific calculation method used is the one described in the study [17]. This study proposes a method for approximating data on a polygonal line with multiple division points. The quadratic function is calculated using the least squares method, one of the famous methods of regression analysis.

3.4. Judging Method

Since the trajectory of the ball was calculated in the processing described so far in two ways, the information of the calculated trajectory is then used to judge the ”in/out” of the batted ball. Two methods of judgement are proposed. The first is to judge from the angle formed by the two linear functions. The second is to judge from the mean square error calculated by the detection point of the ball and the trajectory.

3.4.1. Judgement by the Angle formed by Two Straight Lines

Judgement by the angle of two straight lines is a method that focuses on the change in the traveling direction of the ball due to the ball hitting to the table. Figure 2 shows the outline of the method.

The method uses two linear function out of two calculated trajectories. Figure 2 show a diagram assuming that the ball enters the angle of view from the upper right. First, let the two calculated lines be vectors. As for the start and end points, it is easy to determine from the frame in which the ball is detected that the detection point in the preceding frame is the start point and the detection point in the following frame is the end point. Of these, the vector of the first half divided by time is v_1 and the second half is v_2. At this time, when the angle formed by the two vectors is considered the direction of rotation, there are two ways as shown in Figure 2. When the ball hits the table, the movement changes at the boundary of the touched point. Since the change is upward from the traveling direction of the ball, when this is expressed by two vectors, v_2 takes a v_1 to clockwise angle as shown on the left in Figure 2. On the other hand, if the table is not touched, the ball draws a parabola. This change is downward from the traveling direction of the ball, and when expressed in terms of two vectors, it takes an angle such that v_2 is counterclockwise relative to v_1 as shown on the right in Figure 2. Thus, the ”in/out” of a batted ball can be judged from the pattern of angles formed by the two straight lines.

3.4.2. Judgement by the Errors in Two Trajectories

Judgement by the errors in two trajectories is a method to judge by the value of the error of each trajectory calculated by the detected position of the batted ball. We used the mean square error (MSE) shown in equation (3) as the error.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{3}$$

The mean square error is used to see the difference between the function value and the actual value for a function. n is the number of detected balls. $\hat{y}_i$ and y_i are the y-coordinate of the function value and the center of the ball, respectively. By using these values, we confirm how well the calculated trajectory correctly expresses the movement of the detection point.

This time, we calculate two trajectories that expressed by two linear and quadratic functions, and it is clear that the error of the quadratic function, which is a parabola, decreases when

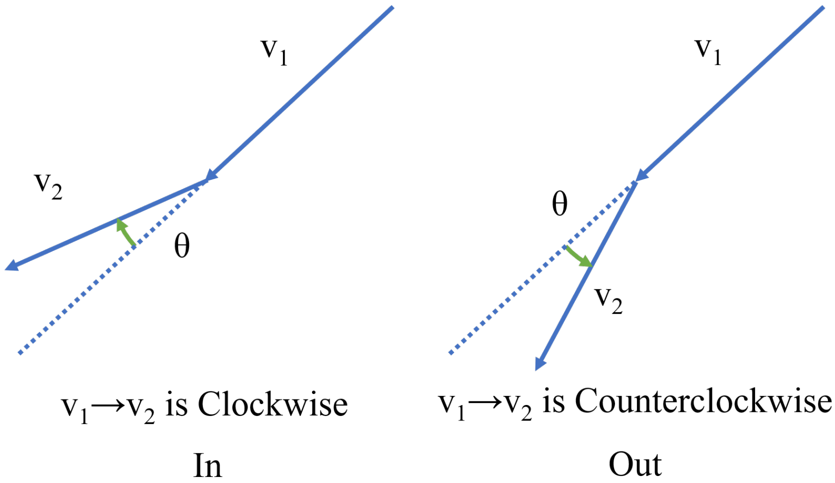

Figure 2: Judgement by the Angle formed by Two Straight Lines

Table 1
Relationship between the Values of Errors in the Trajectories

	Two Linear Functions	Quadratic Function
in	small	large
out	large	small

the batted ball is ”out.” On the other hand, it is predicted that the error is smaller when the trajectory by two linear is applied, because when the ball hits the table, the movement changes around the contact point. In summary, the values of the error are shown in Table 1.

Judgement of the ”in/out” of the batted ball is made from the relation of the value of the error. Specifically, if the mean square error of the trajectory by the two linear functions and the center point of the ball is smaller than that of the trajectory by the quadratic function, the batted ball is judged as ”in.” If it is large, the batted ball is judged as ”out.”

3.5. Experiments

3.5.1. Details of Experiments

We have experiments to evaluate how accurately the judging methods can correctly judge ”in/out.” For ball photography, we use RX0 [18]. This camera can record video at 960 fps, which means it can accurately capture table tennis ball even though it moves fast. The resolution is 800 px×270 px, resized to 1920 px×1080 px and output. The camera is positioned as an extension of the end line so that the edge of the table can be seen, and the distance is about 0.1m from the edge of the table.

The amount of videos are shown in Table 2. The total number of videos is 44, of which 29 are ”in” and 15 are ”out.” In, 14 bounce on the plane of the table and 15 hit the edge.

Table 2
The Number of Videos

	In		Out	Total
	Plane	Edge		
Number of data	14	15	15	44

Table 3
Judgement Classification in Confusion Matrices

		Judgement Result	
		In	Out
Correct Result	In	TP	FN
	Out	FP	TN

Table 4
Judgement Result by the Angle Formed by Two Straight Lines

		Judgement Result	
		In	Out
Correct Result	In	27	2
	Out	0	15

The accuracy is evaluated using a confusion matrix. Here, the positive is matched with judging ”in” and the negative with judging ”out.” An example of the table is shown in Table 3. The confusion matrix classifies the data into four types, TP, FN, FP and TN, as shown in Table 3. TP (True Positive) is defined as data for which a positive judgement is made on positive data, that is, data for which an ”in” is correctly judged as an ”in.” Similarly, TN (True Negative) is defined as data for which an ”out” is correctly judged as an ”out.” FN (False Negative) and FP (False Positive) are data for which positive/negative data are judged as negative/positive, respectively, and in this case, data for which an ”in/out” data is incorrectly judged as ”out/in.” In other words, the greater the number of TP and TN, the higher the accuracy of the judgement, and the Accuracy is expressed by Equation (4).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

3.5.2. Experimental Results

First, the confusion matrix of the results of judging by the angle formed by the two straight lines is shown in Table 4. From Table 4, Accuracy was 95.5% calculated by using Equation (4) . The data includes batted balls such as hits on a plane and simply ”out,” which can be clearly judged by human eyes. Therefore, it is reasonable that the judgement was made with such a high accuracy rate. On the other hand, two results of false positives in which a batted ball, which was originally ”in,” was judged as ”out.” These trajectories are shown in Figure 3.

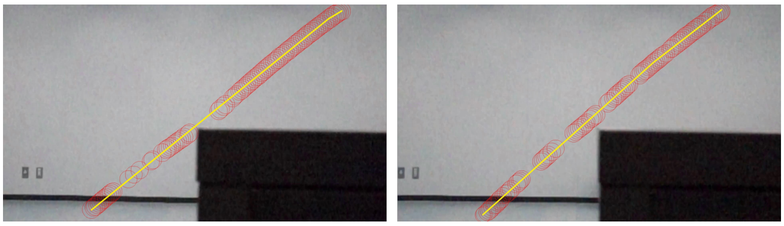

Figure 3: Examples of Edge Ball that was not Correctly Judged

Table 5
Judgement Result by the Errors in Two Trajectories

		Judgement Result	
		In	Out
Correct Result	In	29	0
	Out	0	15

Table 6
Mean Square Error of each Trajectory in Figure 3

	Mean Square Error	
	Two Linear Functions	Quadratic Function
Left in Figure 3	26.26	28.04
Right in Figure 3	10.58	14.79

Those shown by red circles are the results of shaping the detected balls into circles, and yellow lines are the trajectories calculated based on the center coordinates of the red circles. In the judgement, the calculations are performed using the center of the red circles and the function values of the trajectories shown in yellow. The two examples shown in Figure 3 are both edge balls. The trajectories shown on the right in Figure 2 were calculated for these two trajectories, and they were judged to be ”out” even though they hit the edge. The reason for these results is considered to be that the change in the motion of the ball caused by gravity was judged to be greater than the change caused by hitting the edge. In both cases, the breakpoint of the trajectory exists before the ball hit the table. When the value of each angle was checked, the left in Figure 3 was 8.13 degrees and the right was 4.03 degrees. From this, it is considered that, even for an edge ball, if the change in angle due to contact with the edge is small, there are cases where the judgement is not carried out correctly in this method.

Next, the confusion matrix of the results of judging by the errors in two trajectories is shown in Table 5. From table 5, it can be seen that all the videos taken were judged correctly, including the racy edge ball which was not judged correctly by the angle formed by two straight lines. In addition, the mean square error is shown in Table 6. Although the difference in error is small in both cases, the error when the trajectory is calculated by the two linear functions is smaller, so it can be seen that it was correctly judged as ”in.”

Table 7
The Number of Videos

Distance [m]	In		Out	Total
	Plane	Edge		
1.0	11	8	10	29
1.5	11	10	12	33
2.0	16	12	9	37

4. Consideration of the Distance between the Table Tennis Table and the Camera

In the experiment in Section 3, the judgement was made based on the images shown in Figures 3, and the distance between the camera and the table tennis table was about 0.1 m. However, it is not practical to set the camera in such a position during an actual match because it would disturb the match. Therefore, we set the camera in a position away from the table tennis table and considered how the judgement result in the video taken from it is affected. We also examine what kind of processing should be done to reduce the effect.

4.1. Relationship between the Distance and the Judgement

The farther away the camera is from the table, the more extensive the ball detection will be. Because the trajectory is calculated from a wider range of ball detection points, the calculation method used in Section 3 may calculate trajectories with large errors. Therefore, by narrowing the processing range, we express the trajectory at local ball detection points, and try to obtain a trajectory equivalent to video used in the experiment in Section 3.

The range are based on the size of the ball in the video. When the distance between the stand and the camera is 0.1 m, the radius of the detected ball is about 50 px. Since the position of the camera and the injection location of the ball and the shooting point are constant, this value does not change significantly. The processing range is limited so that the ratio of the radius of the ball to the size of the original video. Let r be the radius of the ball when the distance between the stand and the camera is separated, w be the width of the processing range of the video at this time, and h be the height, which can be determined as shown in Equations (5) and (6).

$$w = 1920 \times \frac{r}{50} \tag{5}$$

$$h = 1080 \times \frac{r}{50} \tag{6}$$

4.2. Experiments

In the experiment, we prepare videos taken with the table and camera separated by 1.0 m, 1.5 m, and 2.0 m, and evaluate the results of ”in/out” judgement from the original videos and when the range is limited. First, the number of videos taken is shown in Table 7.

Table 8
Judgement Result by the Angle Formed by Two Straight Lines

Distance [m]	Range Limitation	Accuracy [%]
1.0	no	93.1
	yes	96.6
1.5	no	90.9
	yes	100.0
2.0	no	83.8
	yes	97.3

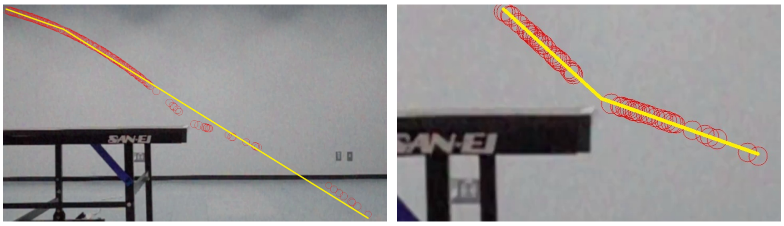

Figure 4: The Example Judged Correctly with Range Limitation

Table 9
Judgement Result by the Errors in Two Trajectories

Distance [m]	Range Limitation	Accuracy [%]
1.0	no	100.0
	yes	100.0
1.5	no	100.0
	yes	100.0
2.0	no	100.0
	yes	100.0

Next, the results of ”in/out” judgement are described. Table 8 shows the accuracy of the judgement by the angle formed by two straight lines with changing the distance and the processing range. When the processing range was not limited, the accuracy decreased as the distance between the table and the camera increased, but it was increased by limiting the processing range for videos shot from any distance. An example in which the judgement changed to correct by limiting the processing range is shown in Figure 4. The left in Figure 4 is the result of calculating the trajectory from the original video, the right is the result that the judgement was changed from ”out” to ”in” by limiting the processing range. By limiting the processing range, it can be seen that the trajectory was calculated from the detection point only near the edge of the table, and the accurate trajectory was calculated.

Next, Table 9 shows the accuracy of the judgement by the errors in two trajectories with changing the distance and the processing range. From Table 9 shows, the judgement was made correctly for all videos with and without range restrictions. Therefore, it is possible to judge correctly without limiting the processing range in the judgement by the errors in two trajectories. In the comparison of each method, it is concluded that the judgement by the errors in two trajectories shows good accuracy and is excellent even if the distance between the table and the camera is separated.

5. Conclusion

In this paper, we propose a method for judging the ”in/out” of batted balls, including edge balls. In the proposed method, two kinds of trajectories are calculated from balls detected by object detection, and the ”in/out” of batted balls is judged by the information of the trajectories. There are two proposed methods, and in the first method, judging by the angle formed by two straight lines, it is possible to judge the ”in/out” of an edge ball as much as the human eye can see. In the other method, judging by the error of two trajectories, it is correctly judged even for the racy edge ball which is difficult to judge visually. In the proposed method, it can be said that judging by the error of two trajectories showed better results regardless of the distance between the camera and the table.

However, some problems remain to be solved for practical application. For, examples, include ensuring real-time performance by improving ball detection speed and considering that a part of a batted ball can not be seen by a player. In the future, we aim to find and solve these problems.

References

[1] Sony group portal | news release | takkyu no kokunai ri-gu ”t ri-gu” ni bideo hanntei shisutemu wo teikyou [sony group portal | news release | video judgement system for japanese table tennis league ”t-league”], 2019. URL: https://www.sony.com/ja/SonyInfo/News/Press/201908/19-078/, last accessed:2023/04/20.

[2] K. Lu, C. Liu, H. Zou, Y. Wang, G. Wang, D. Li, K. Fan, W. Yang, L. Dong, R. Sha, D. Li, Self-powered acceleration sensors arrayed by swarm intelligence for table tennis umpiring system, PLOS ONE 17 (2022) 1–23. URL: https://doi.org/10.1371/journal.pone.0272632. doi:10.1371/journal.pone.0272632.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer, 2016, pp. 21–37.

[4] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 28, Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[5] C.-H. Hung, A study of automatic and real-time table tennis fault serve detection system, Sports 6 (2018). URL: https://www.mdpi.com/2075-4663/6/4/158. doi:10.3390/sports6040158.

[6] S. Gómez-González, Y. Nemmour, B. Schölkopf, J. Peters, Reliable real time ball tracking for robot table tennis, Robotics 8 (2019) 90.
[7] H.-I. Lin, Z. Yu, Y.-C. Huang, Ball tracking and trajectory prediction for table-tennis robots, Sensors (Basel, Switzerland) 20 (2020).
[8] Z. Zhang, D. Xu, M. Tan, Visual measurement and prediction of ball trajectory for table tennis robot, IEEE Transactions on Instrumentation and Measurement 59 (2010) 3195–3205. doi:10.1109/TIM.2010.2047128.
[9] K. Zhang, Z. Cao, J. Liu, Z. Fang, M. Tan, Real-time visual measurement with opponent hitting behavior for table tennis robot, IEEE Transactions on Instrumentation and Measurement 67 (2018) 811–820. doi:10.1109/TIM.2017.2789139.
[10] F. Qiao, Application of deep learning in automatic detection of technical and tactical indicators of table tennis, PLOS ONE 16 (2021) 1–16. URL: https://doi.org/10.1371/journal.pone.0245259. doi:10.1371/journal.pone.0245259.
[11] C.-H. Hung, Image judgment auxiliary system for table tennis umpiring under low light conditions, Smart Science 7 (2019) 39–46. URL: https://doi.org/10.1080/23080477.2018.1536912. doi:10.1080/23080477.2018.1536912. arXiv:https://doi.org/10.1080/23080477.2018.1536912.
[12] R. Voeikov, N. Falaleev, R. Baikulov, Ttnet: Real-time temporal and spatial video analysis of table tennis, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 3866–3874. doi:10.1109/CVPRW50498.2020.00450.
[13] Models and pre-trained weights - torchvision main documentation, 2017. URL: https://pytorch.org/vision/main/models.html, last accessed:2023/04/20.
[14] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. doi:10.1109/ICCV.2017.322.
[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 740–755.
[16] kouzoukaiseki to keijou delisukuriputa [structural analysis and shape descriptors opencv v2.1 documentation], 2010. URL: http://opencv.jp/opencv-2.1/py/structural_analysis_and_shape_descriptors.html, last accessed:2023/04/20.
[17] Y. Otsuka, M. Yoshihara, 1 naishi 2 no sekkyokuten wo motsu oresen moderu no atehame [fitting a polyline model with one or two inflection points], [Applied Statistics] Ouyoutoukeigaku 5 (1976) 29–39. doi:10.5023/jappstat.5.29.
[18] Rx0(dsc-rx0) | dejitaru suchiru kamera cyber-shot saiba- shotto | sony [rx0(dsc-rx0) | degital still camera cyber-shot | sony], 2017. URL: https://www.sony.jp/cyber-shot/products/DSC-RX0/, last accessed:2023/04/20.