Precise Ball Detection in Table Tennis Games Using Deep Learning and Stereo Vision

Guang Liang Yang, Minh Nguyen, Xuejun Li, Wei Qi Yan
Department of Computer and Information Sciences
Auckland University of Technology, 1010 New Zealand

ABSTRACT

In this book chapter, we present a novel approach to real-time detection and trajectory tracking of balls in table tennis using computer vision and deep learning, specifically YOLO models and binocular stereo vision. By leveraging a multicamera system and precise calibration of the standard table of table tennis games, we accurately reconstruct the ball in stereo vision, enabling more accurate predictions of its speed, trajectory, and landing points. By combining the rapid detection capabilities of YOLO models and a "third eye" method, we enhance the model performance in real scenes. Our experimental results show that the layer of a small object detection in deep nets significantly improves detection accuracy, with the optimized YOLOv10 model achieving a mAP@0.5 of 0.853, notably surpassing the peak values of YOLOv8 (0.778) and YOLOv10n (0.673). Moreover, the integration of pseudo-labels, generated via stereo vision and "third eye" method, expands the dataset with blurred and occluded balls and improves the accuracy of ball detection. Our data augmentation and self-supervised learning method further optimize model performance and significantly enhance the precision score.

Keywords: YOLOv10 ·RT-DETR· Moving balls · Binocular stereo vision· Self-supervised learning · Trajectory prediction · Table tennis · Deep learning

INTRODUCTION

Table tennis has exceptionally technical requirements as a high-speed and precise sport. Especially in modern competitions, real-time tracking of the table tennis ball with GPU and accurate prediction of its trajectory and landing point has become the focus of this topic. The ball has a fast speed, small size, light weight and complex trajectory. Therefore, achieving high-precision detection and trajectory tracking of the balls in real time has become a major challenge in computer vision.

With the rapid development of deep learning, much more research work is devoted to improve small object detection and track accuracy and efficiency. TrackNet, YOLO (i.e., You Only Look Once), and DETR (i.e., Detection Transformer) have made significant progress among these studies. TrackNet focuses on trajectory tracking of ball sports, can input multiple frames, and predict occluded objects. YOLO is widely employed in various scenarios for its fast and efficient target detection; in multiframe motion, DETR can capture the position of the target object through a cross-frame attention mechanism to improve the detection and tracking accuracy of small objects.

In our research project, not only the real-time detection of table tennis ball, but also the trajectory also needs to be accurately tracked to improve the prediction of landing points of table tennis balls. However, the trajectory prediction of a table tennis ball is far beyond simple. Affected by gravity, air resistance, and the spin of the table tennis ball, its trajectory is not a plane parabola curve according to the Magnus effect but a smooth curve in 3D space. Therefore, we introduced stereo vision based on a binocular camera to obtain the 3D location of the ball. Based on these precise 3D coordinates, we track the movement of the table tennis ball in real time and accurately predict its landing point.

However, broadcast videos usually take use of a single camera in table tennis matches. This single-camera setup is primarily intended for viewing purposes, with the camera typically positioned on a higher stand to capture the entire field of view. Although this setting improves audience experience, it also leads to the loss of depth information, which challenges the prediction of table tennis balls. In addition, in real scenarios, the camera position is often variable, and a constant viewing angle and distance cannot be guaranteed, which further increases the difficulties.

Thus, in this project, we employed a standard table for table tennis games as a reference target and calibrated three cameras to reconstruct a complete 3D scene. Throughout this multicamera stereoscopic vision setting, we accurately reconstruct the 3D trajectory of table tennis ball, thereby improve the accuracy of ball speed measurement and landing point statistics.

In addition to use multiple cameras to locate the ball, we take use of information from the multicamera system to complement the missing information of another camera, thereby generate further data for algorithm training, which improves the training speed and quality of our deep learning models.

The structure of this book chapter includes related work, methodology, result analysis, discussion, conclusion, and future work. Firstly, in related work section, we will analyse the existing work, especially how to apply computer vision to the analysis of table tennis games. Next, we will elucidate the purpose and method of our experiments in combination with specific table tennis scenarios and illustrate the customized design of data acquisition and the basic principles of computational methods. Thus, we will compare and analyse our experimental results to explore the advantages and disadvantages of our methods in the ball tracking. Finally, this book chapter will summarize the research results and propound future research directions and possible improvement methods.

Throughout the research work in this book chapter, we aim to promote the development of ball motion analysis in table tennis, especially in real-time ball detection and trajectory prediction.

RELATED WORK

In the field of visual object detection, YOLO (You Only Look Once) and DETR (Detection Transformer) are two representative models, with distinct advantages and application scenarios. The YOLO series is renowned for its extremely fast detection speed and high accuracy. It is particularly suitable for applications with real-time requirements, such as autonomous driving, security surveillance, and sports analysis (Cao &Yan, 2022; Chen & Yan, 2024). In table tennis, the ball is small, fast, and follows a complex trajectory. YOLO models excel in quickly locating the ball and demonstrate superior performance in detecting fast-moving visual objects. YOLO models achieve visual object localization and classification in a single forward pass, and the lightweight structure enables it to run on devices with lower hardware requirements, which is highly valuable for real-time analysis in table tennis.

YOLO models have progressively improved small object detection from YOLOv1 to YOLOv10. While YOLOv1 to YOLOv3 focused on real-time performance, their ability to detect small objects was limited. YOLOv4 encloses CSPNet and PANet and improved the accuracy of small object detection. YOLOv5 optimized model structure and data augmentation enhanced speed and performance for small moving objects. YOLOv6 to YOLOv8 further refined deep learning architectures and training strategies, while

YOLOv9 incorporated cutting-edge technologies like Transformers and significantly beefed the accuracy and robustness of small object detection (Yan, 2023).

A team from Tsinghua University proposed YOLOv10, which further enhanced the performance and efficiency of YOLO models. YOLOv10 addressed issues such as dependency on Non-Maximum Suppression (NMS) in postprocessing and insufficient scrutiny of YOLO components, which led to computational redundancy and suboptimal efficiency. Based on the COCO dataset, YOLOv10-S is 18 times faster than RT-DETR-R18, with 2.8 times fewer parameters and floating-point operations. Compared to YOLOv9-C, YOLOv10-B achieved 46% reduction in latency and 25% reduction in parameters while maintaining the same performance (Wang, et al. 2024)

In contrast, DETR (Zhao, et al. 2024) is a transformer-based object detection model that captures global features through a self-attention mechanism. It is particularly suitable for complex scenes. DETR avoids traditional anchor-box mechanism, instead directly predicting bounding box positions and classes. While this approach excels in multiobject detection for large images, its detection speed is relatively slower. Although the improved RT-DETR surpasses YOLOv8 in speed (Zhao, et al. 2024), it does not exceed YOLOv10. In table tennis, the longer inference time of DETR and lower sensitivity to small objects make it less ideal for high-speed applications, where YOLO generally performs better. Thus, YOLO is more suitable for detecting fast-moving small objects in table tennis, while DETR excels in detecting static or slow-moving objects in complex backgrounds.

TrackNet is a model specifically designed for sports scenarios. It is particularly effective in detecting and tracking fast-moving objects like table tennis balls (Huang, et al. 2019). By combining detection and tracking in an end-to-end framework, TrackNet processes video frame sequences and leverages temporal information to predict object positions. This gives TrackNet a significant advantage in capturing the precise position of a fast-moving ball in each frame. However, as a specialized model, it heavily relies on specific training datasets and may not be suitable for more flexible and dynamic environments.

Multiple publications (An & Yan, 2021; Calandre, et al, 2021; Lin, Yu, & Huang, 2020) have showcased the forces affecting a table tennis ball during play. These primarily include gravity, drag force, and Magnus force due to spin, as shown in Equation (1).

$$m \cdot \frac{d^2\overrightarrow{r(t)}}{dt^2} = -m \cdot g \cdot \hat{k} - \frac{1}{2} \cdot C_d \cdot \rho \cdot A \cdot |\vec{v}(t)| \cdot \vec{v}(t) + S \cdot \rho \cdot A \cdot \left(\vec{\omega} \times \vec{v}(t)\right) \tag{1}$$

where ρ represents the air density, which affects the magnitude of aerodynamic forces. ω is the angular velocity of the ball, which generates the Magnus effect due to the ball's spin. $\vec{v}(t)$is the velocity of the ball relative to the air, influencing both the drag and Magnus forces. The term g stands for gravitational acceleration, which pulls the ball downwards, and mmm is the mass of the ball, affecting how much it accelerates in response to the forces. A represents the cross-sectional area of the ball, playing a role in determining the size of the aerodynamic drag and Magnus forces. C_d is the drag coefficient, describing the ball's resistance to air movement based on its shape and surface properties. S is a constant that scales the Magnus force, which is the force resulting from the ball's spin interacting with its velocity. In addition, d refers to the distance or displacement over time, typically represented in the form of derivatives, such as $\frac{d^2\overrightarrow{r(t)}}{dt^2}$, which is the acceleration or second derivative of the position vector $\vec{v}(t)$. This measures how fast the ball's velocity changes over time. Lastly, $\hat{k}$ is a unit vector in the vertical direction, often pointing upwards or downwards depending on the coordinate system, representing the direction of gravitational force. Equation (1) shows the nonlinear dynamic system, with each force varying over time. Therefore, the trajectory of ball is usually not confined to a curve within a single vertical plane.

As analysed (Zhou, Nguyen, & Yan, 2024), the assumption of ball moving within a plane is limited. The study utilized a single camera to detect the moment the ball hits the table. The ball contacts with the table is determined by comparing the sign changes in the *y*-axis coordinates of the ball centre between the previous two frames and the current frame. Furthermore, the ball landing area was estimated by comparing the lower boundary of its bounding box with predefined table areas. However, the coordinates only consist of (x, y) coordinates, lacking depth (z-axis) data, which makes it impossible to determine the ball height or precise 3D position. The method only provides a rough estimate of the ball landing area by comparing overlapping regions without calculating the exact landing point. Moreover, the camera angle influences (x, y) coordinates and table area division in the image. The errors are magnified when the ball is far from the camera or at extreme angles. Thus, camera placement is critical, and table areas must be as prominent as possible in the video frame, which also limits the bounce height of the ball.

The method for calculating ball displacement (Zhou, Nguyen, & Yan, 2024)) has significant limitations. By initializing the *z*-coordinate as a unified value for perspective transformation, 2D image coordinates are mapped to 3D space and the depth information along the z-axis is constant as assumed. This approach only accurately computes the movement in the *x*- and *y*-plane and cannot capture variations in *z*-axis. Although auxiliary cameras from the side were suggested to reduce errors from movement along the optical axis of the primary camera, all calculations are still based on 2D coordinates. This simplification leads to limitations in the results, where the calculated velocity only reflects movement in the image plane rather than the real-world velocity of the ball. As a result, this method may fail to accurately predict the trajectory of a flying ball, especially in scenarios which requires high precision.

The camera calibration process has significant limitations, especially in acquiring extrinsic parameters (Zhou, Nguyen, & Yan, 2024). A chessboard must be brought to the table for calibration, and any changes to the training venue or camera position necessitate recalibration. Using the standard dimensions of table and net to calculate extrinsic parameters requires manual pixel location in images to make automatic calibration impossible. Klette (Klette, 2014) introduced calibration principles, where intrinsic and extrinsic parameters are precisely calculated by minimizing reprojection errors using nonlinear optimization algorithms, such as Levenberg-Marquardt algorithm. The study also explained how epipolar geometry can be employed in stereo vision to determine 3D coordinates through triangulation by matching corresponding points in left and right images. Moreover, the "Third-Eye" method utilizes images and geometric information from two designated cameras (stereo cameras) to estimate the content of a hypothetical third-camera view (Klette, 2014).

The model optimization process incorporates Self-Supervised Learning (SSL), a machine learning method that automatically generates supervision signals from unlabelled data for model training. Unlike supervised learning, SSL does not rely on large amounts of manually labelled data but designs specific tasks (such as image contrast, frame prediction, or occlusion recovery) to let the model extract information from the data itself. This method takes advantage of the inherent structure or attributes of the data to generate pseudo-labels through comparison or prediction and guides the learning process of the model (Yan, 2023).

METHODOLOGY

Dataset

In table tennis games, the complexity and dynamic nature of playing environment makes standardized datasets, such as TTNet and COCO, which are difficult to apply directly to our experimental scenario. While we leverage pretrained models to obtain initial parameters as a basis for model improvement, data collection and image processing must be optimized according to specific requirements. The custom training dataset in this book chapter incorporates geometric principles of binocular stereo vision and self-supervised learning, utilizing a multicamera system to generate incremental data and improve model detection performance.

We employed three cameras, as shown in Fig. 1, with two cameras (C1 and C2) capturing at 60 fps and the third camera (C3) at 30 fps. Due to the lower frame rate of C3, the images of the table tennis ball captured by this camera are more prone to blurry, contributing to the dataset diversity by providing various representations of the ball captured by using a low-speed camera. Data augmentation, such as variations in the ball size, rotation angle, and colour, further enhanced the model generalization capabilities. While annotating the position of the table tennis ball, we primarily recorded its *x*- and *y*-coordinate in the image. Given the high-speed movement of the ball, blurry or trailing may occur in the images. For example, as shown in Fig. 1, the ball moves from right to left and leaves a motion trail, with the red circle indicating the marked coordinate. In these cases, we ensured the accuracy of the coordinates by marking the most recent position of the ball, providing a stable and reliable data input.

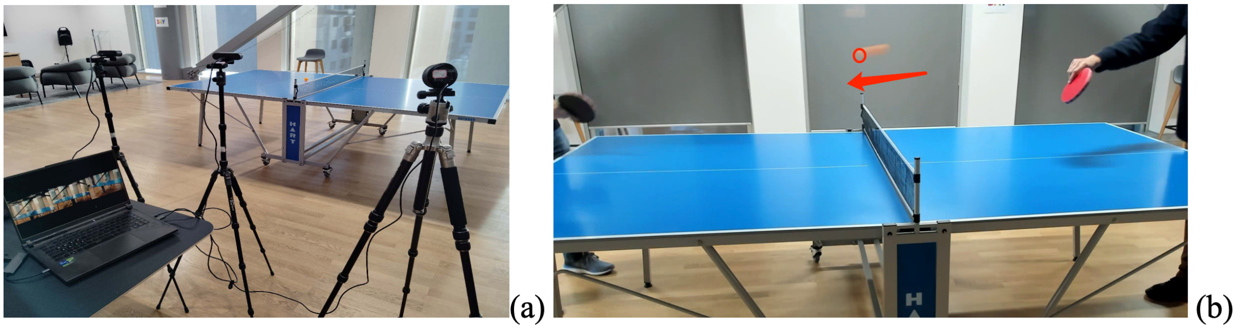

Fig. 1. (a) Multicamera setup for capturing ping-pong balls (b)Annotated the balls with motion blur and trailing effect.

In the context of data augmentation for table tennis, we employed image augmentation methods aimed at improving the robustness of object detection model, particularly for detecting and tracking fast-moving small objects like a table tennis ball.

As shown in **Error! Reference source not found.**, scaling operations was applied by enlarging or shrinking the images, allowing the model to handle objects at different distances. This is crucial in table tennis matches, where players are positioned at varying distances from the ball. Additionally, rotation was employed to randomly rotate images, enabling the model to recognize the trajectory and spin from different angles of balls, which improved its adaptability to changes in ball orientation.

Translation involved randomly shifting the image horizontally, simulating the ball movement across the playing field, and helping the model better respond to the rapid motion of the ball. Brightness adjustment and colour jittering were employed to maintain detection performance under varying lighting conditions and colour changes, addressing the complex lighting environments often in table tennis venues.

Lastly, Gaussian blurring simulated the effects of camera blur or motion blur due to fast movement, which enhanced the model's ability to detect the fast-moving ball even when image quality is degraded.

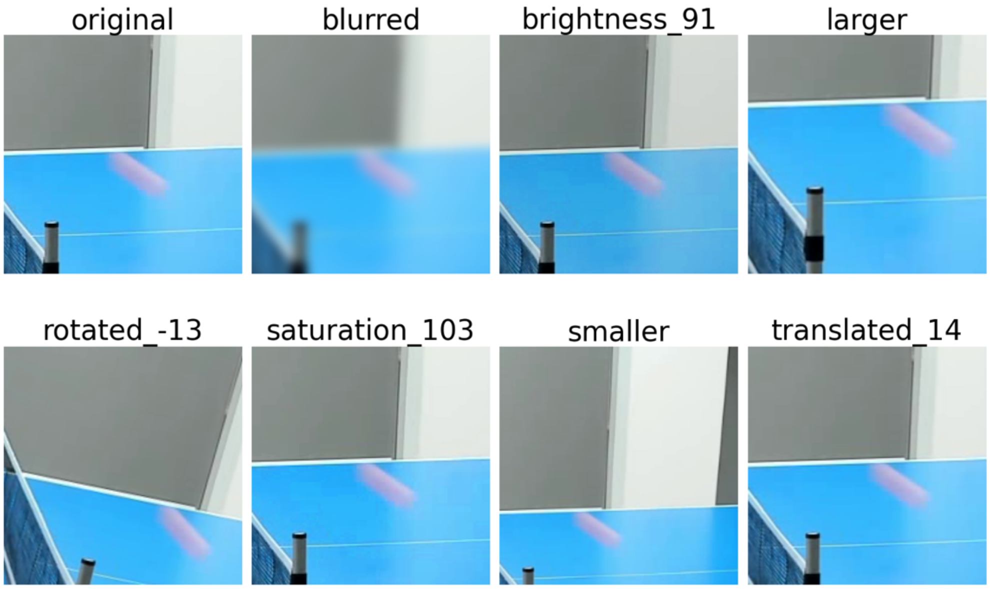

Fig. 2. Data augmentation methods applied to the ball detection in table tennis games.

Throughout this iterative process, the model progressively improves its detection accuracy across all cameras and maintains high precision even in complex game environments if camera frame rates are inconsistent. The combination of a custom dataset based on binocular stereo vision and self-supervised learning not only captures precise motion information of the table tennis ball but also provides robust training data through diverse image augmentation and pseudo-label strategies which ensures the broad applicability of the model in various scenarios.

Modelling

In the modeling phase, we compared the performance of two object detection models, YOLOv10n and YOLOv8 (Wang, et al. 2024), within a real-time setup. The computational environment consists of an NVIDIA GeForce RTX 4090 (24217MiB). The vision system employed three cameras: Two Logitech Brio 4K cameras, each operating at 90fps, and one Razer Kiyo Pro Ultra camera at 60fps. The two Brio 4K cameras were utilized as the primary binocular stereo cameras for calculating the 3D coordinates of the table tennis ball. In contrast, the Razer Kiyo Pro Ultra camera provided auxiliary data to support self-supervised incremental learning, as shown in Fig. 1.

The spatial reference system was established with the far-left corner of the table as the origin. It defined *x*-axis along the width, *y*-axis along the length, and *z*-axis vertically. The table adhered to standard competition dimensions: 1.525m in width, 0.76m in height, and 2.74m in length, with a 0.1525m net height, as shown in Fig. 3. The intrinsic parameters of the cameras were calibrated by using a chessboard pattern, after the system transitioned to determining extrinsic parameters via using nine reference points located on the table, as shown in Fig. 3.

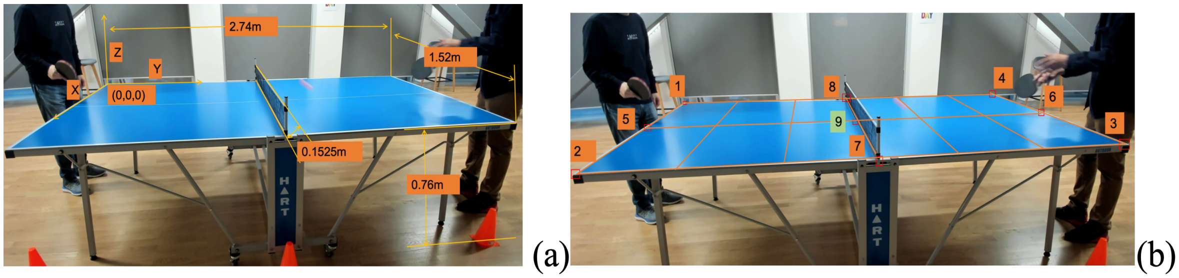

Fig. 3. (a) Standard table dimensions. (b) Camera calibration using table reference points.

Pertaining to calculating instantaneous ball speed, the 3D physical distance between the ball positions in consecutive frames was divided by the time between adjacent frames. Additionally, to maintain consistency with previous experiments, the table was divided into nine equal regions along each side to analyze ball landing spots, as shown in Fig. 3.

In addition, we added a customer-modified YOLOv10 model, which introduced an additional detection layer specifically designed to enhance the model performance on small object detection, as illustrated in **Error! Reference source not found.**. The network was augmented by adding a new detection branch at a lower feature map resolution (P2), which is essential for handling small objects like ping pong balls that are often missed by standard object detection layers. The P2 detection branch is integrated using additional upsampling and concatenation layers, as shown in the figure. This structure allows the model to capture finer spatial details by leveraging high-resolution features from earlier stages in the network. The model achieves better detection accuracy by detecting objects at multiple scales (P2, P3, P4, P5), particularly for small and fast-moving objects. The modifications optimize the trade-off between model size and detection performance, which results in a more robust and precise model for small object detection tasks in dynamic environments.

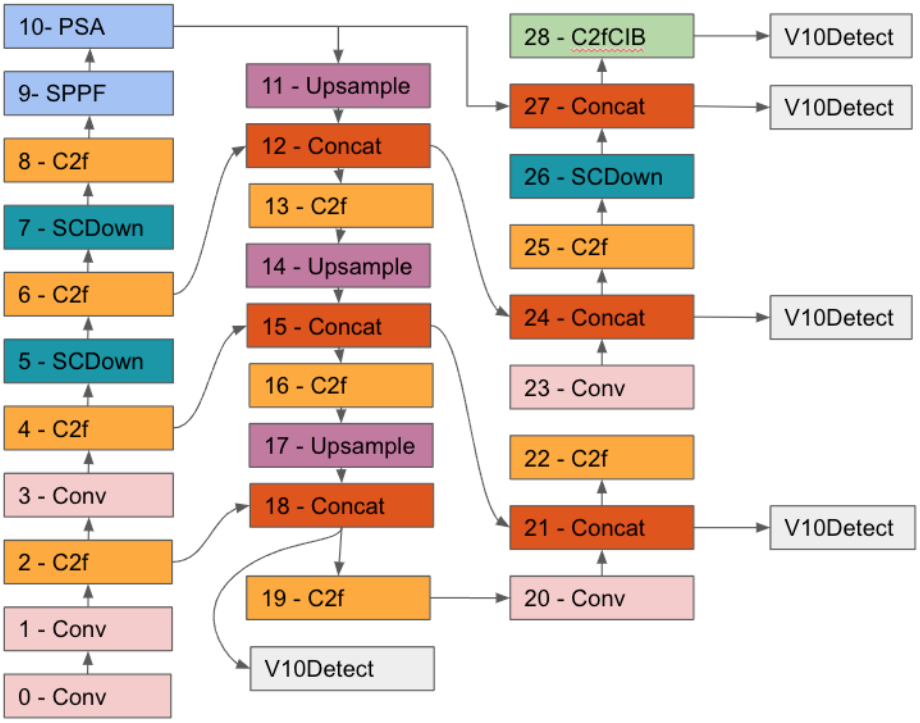

Fig. 4. The architecture of modified YOLO v10 with enhanced small object detection layers.

Method

To adapt YOLO v10 for the specific demands of detecting balls in table tennis games, we implemented a significant architectural modification, as depicted in **Error! Reference source not found.**. A new detection branch was brought in by adding an upsampling layer (17) and a concatenation layer (18) to create a P2-level detection branch targeting small objects. This additional branch enhances the model's capability to detect fine-scale features by preserving the spatial resolution of smaller objects. The network now operates on four detection scales: P2 for small objects, P3 for medium-sized objects, P4 for large objects, and P5 for large objects. The modified YOLOv10 can maintain high detection precision across varying object sizes by incorporating these additional layers. The network diagram demonstrates the flow of information through the backbone, with newly added layers facilitating multi-scale detection. This multiscale feature extraction ensures that the model can detect small, fast-moving objects common in table tennis scenarios.

To accurately obtain the intrinsic parameters of the camera, we utilized classical chessboard calibration method (Klette, 2014). The data from different perspectives were collected by moving, rotating, and tilting the chessboard at various angles. The specific steps included recording videos in front of each camera and extracting over 100 images with clearly defined chessboard corner points for calculation. The intrinsic parameters of camera, including the focal length (f_x,f_y), optical center position (c_x,c_y), and radial and tangential distortion coefficients, were determined based on a physical model. Each 2D point (μ,υ) on the image is mapped to the image plane from the 3D world coordinates (*X*, *Y*, *Z*) by using the projection equations:

$$\binom{\mu}{\upsilon} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \end{pmatrix} \begin{pmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{pmatrix} \tag{1}$$

During the camera calibration, nonlinear distortion is corrected through a radial distortion model through using a polynomial correction of $r^2 = x^2 + y^2$, a tangential distortion model is employed to ensure that the image accurately reflects the projection characteristics of the camera. By selecting over 100 video frames from different angles, the chessboard was exposed to a wide range of perspective variations, while increasing the robustness of parameter estimation.

Firstly, the YOLO model was harnessed to automatically detect eight points on the table of table tennis in the image (Fig. 3). The lines were then drawn between Point 5 and Point 6, as well as Point 7 and Point 8, with the intersection was defined as Point 9. Based on the known physical dimensions of the table (2.74 meters in length and 1.525 meters in width), Point 1 was defined as the origin of the 3D coordinate system. By using the detected positions of the key points in the image, we compute the 3D coordinates for all key points.

This method replaces the traditional chessboard calibration and provides sufficient corner points for extrinsic calibration of the camera. Through using these corner points, we obtain the external parameter matrices of the three cameras, including the camera rotation matrix and translation vector. Precisely, the external parameter matrix is calculated by using Equation (3),

$$[R|t] = K^{-1} \cdot P \tag{2}$$

where *R* is the rotation matrix, *t* is the translation vector, *K* is the camera intrinsic matrix, and *P* is the camera projection matrix.

The object detection model by YOLO models is employed to detect the ball in table tennis games in the images captured by cameras C1 and C2. YOLO model quickly and accurately identifies visual objects in the images and provides the 2D-pixel coordinates of the table tennis ball for both cameras. With the intrinsic and extrinsic parameters of the two cameras, along with the principles of epipolar geometry and binocular stereo vision, the 3D coordinates of the ball are calculated.

Epipolar geometry is one of the core principles of binocular stereo vision. Cameras C1 and C2 each have intrinsic and extrinsic parameters used to transform 2D pixel coordinates in the image into 3D space coordinates. Through the constraints of epipolar geometry, every point detected in C1 corresponds to an epipolar line in C2, thereby reducing the search area in the C2 image. The 3D coordinates of the ball are accurately calculated by determining the intersection of two rays, and this 3D point is then projected onto the image plane of a third camera, C3, using intrinsic and extrinsic parameters to recompute the coordinates (Klette, 2014), where (*X*′, *Y*′, *Z*′) represents the coordinates of the ball in the coordinate system of camera C3, derived from the world coordinates using the extrinsic parameters of camera C3:

$$(\mathrm{X}', \mathrm{Y}', \mathrm{Z}') = R_3 \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T_3 \tag{3}$$

While calculating the speed of the table tennis ball, we determine its velocity in three dimensions by using the time difference between two consecutive frames and the corresponding 3D coordinates. Assuming that 3D coordinates of the ball in frames *i* and *i*+1 are $(X_i, Y_i, Z_i\ )$ and $(X_{i+1}, Y_{i+1}, Z_{i+1})$ respectively, the velocity of ball in each dimension is,

$$V_x = \frac{X_{i+1} - X_i}{\Delta t}, V_y = \frac{Y_{i+1} - Y_i}{\Delta t}, V_z = \frac{Z_{i+1} - Z_i}{\Delta t}, \tag{4}$$

Hence, we obtain the velocity vector of the ball in each dimension. The total speed *V* of the ball can then be derived by calculating the Euclidean norm of these components:

$$V = \sqrt{V_x^2 + V_y^2 + V_z^2} \tag{5}$$

We simplify the process by focusing on the vertical (*z*-coordinate) changes. Specifically, as the ball moves in the air, its *z*-value (vertical coordinate) is influenced by gravity and typically decreases until reaching the minimum point, after which it begins to increase. By detecting the change in the z-coordinate, we identify the lowest point, which represents the landing point.

Mathematically, we define motion of the ball as a discrete time series of 3D points (X_i, Y_i, Z_i), where Z_i is the height of the ball at frame iii. If at any time, the condition $Z_{i-1} > Z_i\ < Z_{i+1}$ is satisfied, then Z_i is a local minimum, indicating that the ball has reached its lowest point, which corresponds to the landing point. At this moment, the landing point of the ball can be approximated by the coordinates (X_i, Y_i, Z_i) of that frame.

$$BP = \begin{cases} Not\ bounce\ point, & if\ Z_{i-1} \le Z_i\ or\ Z_{i+1} \le Z_i \\ Bounce\ point, & if\ Z_{i-1} > Z_i\ < Z_{i+1} \end{cases} \tag{6}$$

By utilizing a simple extremum detection, this method avoids the complexity of parabolic curve fitting and force modelling of the table tennis ball as shown in Equation (6), thereby providing a fast and efficient way to determine the landing point of the ball.

RESULT ANALYSIS

Fig. 5 demonstrates the comparative performance of three models—YOLOv8, YOLOv10n, and our YOLOv10 model—across 500 epochs, specifically focusing on detecting ping pong balls. Our YOLOv10 model, as mentioned in the modelling section, incorporates a small object detection layer to enhance its ability to detect smaller objects. The results show that this modification significantly improves the model's detection performance. Regarding mAP@0.5, our YOLOv10 model achieves a peak value of 0.853 around the 400th epoch, surpassing YOLOv8 and YOLOv10n, which reach maximum values of 0.778 and 0.673, respectively. This increase in performance demonstrates the effectiveness of the small object detection layer in improving the model's capacity to accurately detect small, fast-moving objects such as ping pong balls.

Furthermore, in terms of F1 Score, our YOLOv10 model consistently outperforms the other models, reaching a maximum of 0.786, while YOLOv8 and YOLOv10n reach only 0.664 and 0.596, respectively. The higher F1 Score suggests a better balance between precision and recall, indicating that our YOLOv10 model detects small objects more accurately and reduces false positives. These improvements are significant for ping pong ball detection, where small object precision is crucial. These results confirm that integrating a small object detection layer into the model significantly enhances its detection accuracy and generalization ability, especially for small objects, and allows for faster convergence and improved overall performance compared to the original models.

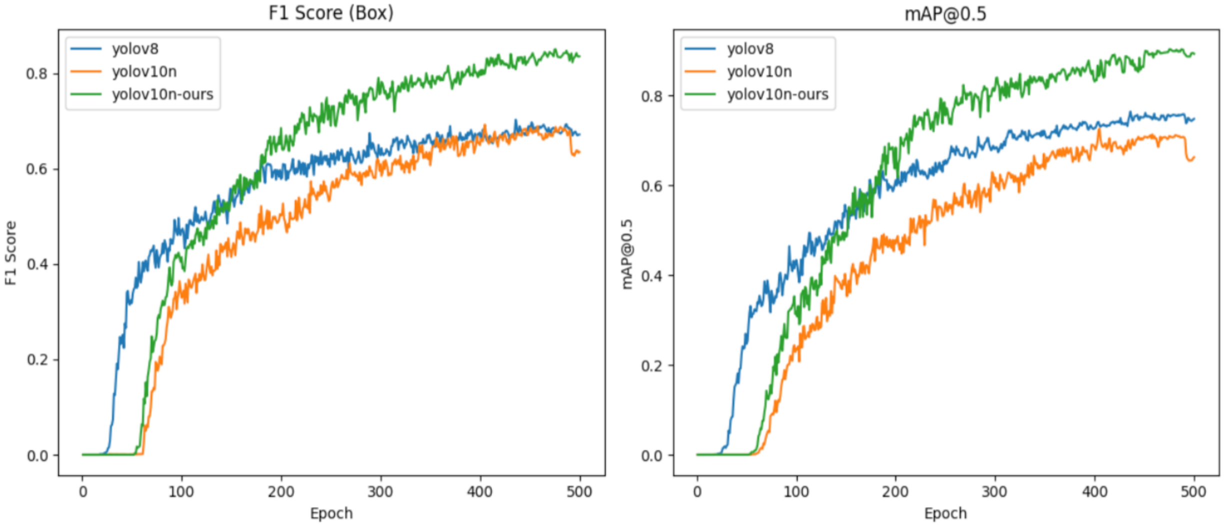

Fig. 5. The impact of incremental training data on YOLOv10 and RT-DETR.

Fig. 6 presents the final output of the table tennis ball detection using deep learning. On the left, the real-time outcomes are from camera C1 and camera C2, while on the right, it shows the statistical data. The statistical data includes the speed of the ball and the probability distribution of its landing spots. In Fig. 6, we detected the speed of the ball is 0.83 km/h, with the landing points mainly concentrated in the L22 and L21 regions on the left. Additionally, it detects a 100% hit rate in the R02 region on the right. This result demonstrates the ability of the system to effectively capture the movement trajectory of the ball and calculate its landing points and probability distribution.

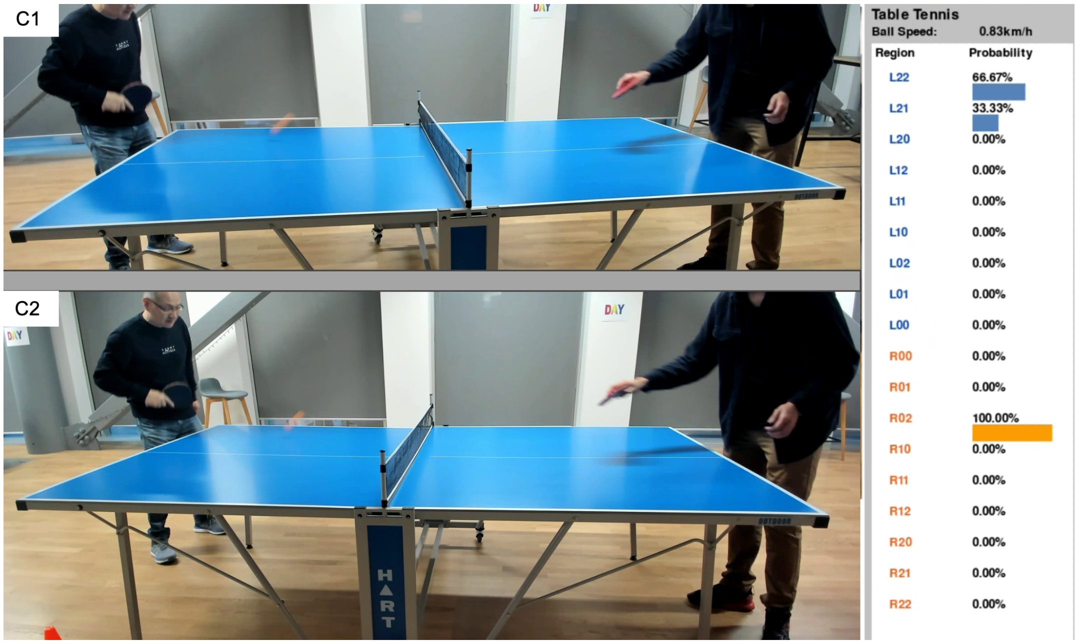

Table Tennis
Ball Speed: 0.83km/h

Region	Probability
L22	66.67%
L21	33.33%
L20	0.00%
L12	0.00%
L11	0.00%
L10	0.00%
L02	0.00%
L01	0.00%
L00	0.00%
R00	0.00%
R01	0.00%
R02	100.00%
R10	0.00%
R11	0.00%
R12	0.00%
R20	0.00%
R21	0.00%
R22	0.00%

Fig. 6. The interface of real-time analysis of table tennis matches.

CONCLUSION

In this book chapter, we explore the application of computer vision, particularly YOLO and binocular stereo vision, for real-time detection and trajectory tracking of balls in table tennis games. By utilizing a multi-camera system calibrated to the dimensions of a standard table tennis table, in this project, we achieved accurate motion of the ball, significantly enhancing the accuracy of both speed estimation and landing point prediction. The integration of the fast detection capabilities of YOLO models with the "third eye" self-supervised learning method allows for incremental model improvements and robust detection, even in complex environments. Our experimental results demonstrate that incorporating a small object detection layer into the model substantially boosts detection accuracy and generalization, particularly for small objects like balls in table tennis, while enabling faster convergence and overall improved performance, most notably reflected in the increased mean average precision (mAP) scores.

Despite the high efficiency of YOLOv10-s, challenges persist in detecting pixel-level key points in more detailed scenes. To overcome this, we integrate classic computer vision techniques, such as Hough transform and contour detection, to accurately compute key points on the table, further enhancing the precision of ball detection. In future, we will work for the reliability in uncontrolled environments for sports as well as general AI related to model transparency, generalizability, and explainability.

REFERENCES

An, N., Yan, W. (2021) Multitarget tracking using Siamese neural networks. ACM Transactions on Multimedia Computing, Communications and Applications.

Bian, J., Li, X., Wang, T., Wang, Q., Huang, J., Liu, C., Zhao, J., Lu, F., Dou, D., & Xiong, H. (2024). P2ANet: A large-scale benchmark for dense action detection from table tennis match broadcasting videos. ACM Trans. Multimedia Comput. Commun. Appl., 20(4), 118:1-118:23.

Calandre, J., Péteri, R., Mascarilla, L., & Tremblais, B. (2021). Extraction and analysis of 3D kinematic parameters of table tennis ball from a single camera. *International Conference on Pattern Recognition (ICPR)*, 9468–9475.

Cao, X. and Yan, W. (2022) Pose estimation for swimmers in video surveillance. Multimedia Tools and Applications, Springer.

Chen, Z., Yan, W. (2024) Real-time pose recognition for billiard player using deep learning. Deep Learning, Reinforcement Learning and the Rise of Intelligent Systems, IGI Global.

Dong, K., Yan, W. (2024) Player performance analysis in table tennis through human action recognition. Computers, 13(12), 332.

Gao, X., Nguyen, M., Yan, W. (2024) HFM-YOLO: A novel lightweight and high-speed object detection model. Optimization, Machine Learning, and Fuzzy Logic: Theory, Algorithms, and Applications (Chapter 16). IGI Global.

Gowdra, N., Sinha, R., MacDonell, S., Yan, W. (2021) Maximum Categorical Cross Entropy (MCCE): A noise-robust alternative loss function to mitigate racial bias in Convolutional Neural Networks (CNNs) by reducing overfitting. Pattern Recognition.

Herrera, A., Beck, A., Bell, D., Miller, P., Wu, Q., Yan, W. (2008) Behavior analysis and prediction in image sequences using rough sets. International Machine Vision and Image Processing Conference (pp.71-76)

Huang, Y.-C., Liao, I.-N., Chen, C.-H., İk, T.-U., & Peng, W.-C. (2019). TrackNet: A deep learning network for tracking high-speed and tiny objects in sports applications. *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 1–8.

Klette, R. (2014). Concise Computer Vision: An Introduction into Theory and Algorithms. Springer London.

Li, F., Zhang, Y., Yan, W., Klette, R. (2016) Adaptive and compressive target tracking based on feature point matching. International Conference on Pattern Recognition (ICPR), (pp.2734-2739).

Liang, C., Lu, J., Yan, W. (2022) Human action recognition from digital videos based on deep learning. ACM ICCCV

Liang, C., Yan, W. (2024) Human action recognition based on YOLOv7. Deep Learning, Reinforcement Learning, and the Rise of Intelligent Systems. IGI Global.

Lin, H.-I., Yu, Z., & Huang, Y.-C. (2020). Ball tracking and trajectory prediction for table-tennis robots. Sensors, 20(2).

Liu, C., Hayakawa, Y., & Nakashima, A. (2012). An on-line algorithm for measuring the translational and rotational velocities of a table tennis ball. SICE Journal of Control, Measurement, and System Integration, 5, 233–241.

Liu, C., Yan, W. (2020) Gait recognition using deep learning. Handbook of Research on Multimedia Cyber Security (pp.214-226)

Liu, Y., Nand, P., Hossain, A., Nguyen, M., Yan, W. (2023) Sign language recognition from digital videos using feature pyramid network with detection Transformer. Multimedia Tools and Applications.

Lu, J., Shen, J., Yan, W., Boris, B. (2017) An empirical study for human behaviors analysis. International Journal of Digital Crime and Forensics 9 (3), 11-17.

Lu, J., Nguyen, M., Yan, W. (2020) Comparative evaluations of human behavior recognition using deep learning. Handbook of Research on Multimedia Cyber Security, 176-189.

Lu, J., Nguyen, M., Yan, W. (2020) Human behavior recognition using deep learning. International Conference on Image and Vision Computing New Zealand.

Lu, J., Nguyen, M., Yan, W. (2021) Sign language recognition from digital videos using deep learning methods. International Symposium on Geometry and Vision.

Luo, Z., Nguyen, M., Yan, W. (2021) Sailboat detection based on automated search attention mechanism and deep learning models. International Conference on Image and Vision Computing New Zealand.

Luo, Z., Nguyen, M., Yan, W. (2022) Kayak and sailboat detection based on the improved YOLO with Transformer. ACM ICCCV.

Nasution, U., Nasution, M. A. H., Habibi, M. I., Tahira, W. L. A., & Ridoh, M. (2024). Analysis of the development of regulations and policies in the world of table tennis: A literature study approach. Journal Coaching Education Sports, 5(1), 25–32.

Pan, C., Yan, W. (2018) A learning-based positive feedback in salient object detection. International Conference on Image and Vision Computing New Zealand.

Pan, C., Yan, W. (2020) Object detection based on saturation of visual perception. Multimedia Tools and Applications, 79 (27-28), 19925-19944.

Pan, C., Liu, J., Yan, W., Zhou, Y. (2021) Salient object detection based on visual perceptual saturation and two-stream hybrid networks. IEEE Transactions on Image Processing.

Poliakov, A., Marraud, D., Reithler, L., & Chatain, C. (2010). Physics based 3D ball tracking for tennis videos. International Workshop on Content Based Multimedia Indexing (CBMI), 1–6.

Qi, J., Nguyen, M., Yan, W. (2022) Small visual object detection in smart waste classification using Transformers with deep learning. International Conference on Image and Vision Computing New Zealand (IVCNZ).

Qi, J., Nguyen, M., Yan, W. (2023) CISO: Co-iteration semi-supervised learning for visual object detection. Multimedia Tools and Applications

Shen, J., Yan, W., Miller, P., Zhou, H. (2010) Human localization in a cluttered space using multiple cameras. IEEE International Conference on Advanced Video and Signal Based Surveillance.

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). YOLOv10: Real-Time End-to-End Object Detection (No. arXiv:2405.14458).

Wang, X., Yan, W. (2019) Human gait recognition based on self-adaptive hidden Markov model. IEEE/ACM Transactions on Biology and Bioinformatics.

Wang, X., Yan, W. (2019) Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term memory. International Journal of Neural Systems.

Wang, X., Yan, W. (2020) Non-local gait feature extraction and human identification. Springer Multimedia Tools and Applications.

Wang, X., Yan, W. (2020) Cross-view gait recognition through ensemble learning. Neural Computing and Applications 32 (11), 7275-7287.

Wang, X., Yan, W. (2022) Human identification based on gait manifold. Applied Intelligence.

Wang, Y. (2021) Colorizing Grayscale CT Images of Human Lung Using Deep Learning. Master’s Thesis, Auckland University of Technology, New Zealand.

Wang, Y., Yan, W. (2022) Colorising grayscale CT images of human lungs using deep learning methods. Springer Multimedia Tools and Applications.

Yan, W. (2019). Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. Springer.

Yan, W. (2023) Computational methods for Deep Learning: Theory, Algorithms, and Implementations. Springer, Singapore.

Yang, B., Yan, W. (2024) Real-time billiard shot stability detection based on YOLOv8. Deep Learning, Reinforcement Learning, and the Rise of Intelligent Systems, pp.159-172, Chapter 8, IGI Global.

Yang, G. L., Nguyen, M., Yan, W. Q., & Li, X. J. (2025). Foul detection for table tennis serves using deep learning. Electronics, 14(1).

Yu, Z., Yan, W. (2020) Human action recognition using deep learning methods. International Conference on Image and Vision Computing New Zealand.

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024). DETRs beat YOLOs on real-time object detection (No. arXiv:2304.08069).

Zhou, H., Nguyen, M., & Yan, W. Q. (2023). Computational analysis of table tennis matches from real-time videos using deep learning. In Pacific-Rim Symposium on Image and Video Technology (pp. 69-81). Singapore: Springer Nature Singapore.

Zheng, K., Yan, Q., Nand, P. (2017) Video dynamics detection using deep neural networks. IEEE Transactions on Emerging Topics in Computational Intelligence.

Zhu, W., Peng, B., Yan, W. (2024) Dual knowledge distillation on multiview pseudo labels for unsupervised person re-identification. IEEE Transactions on Multimedia, 26 (7359 - 7371).

Zhu, Y., Peng, B., Yan, W. (2022) Ski fall detection from digital images using deep learning. ACM ICCCV