Article

Table Tennis Serve Foul Detection Leveraging 3D Ball Trajectories and Deep Learning

GuangLiang Yang \*, Wei Qi Yan\*

Auckland University of Technology, Auckland 1010 New Zealand

|  |
| --- |
| **Citation:** To be added by editorial staff during production.  Academic Editor: Firstname Lastname  Received: date  Revised: date  Accepted: date  Published: date  A grey and black sign with a person in a circle  Description automatically generated  **Copyright:** © 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/). |

**\*** Correspondence: fdp5284@autuni.ac.nz, weiqi.yan@aut.ac.nz

**Abstract:** Detecting serve fouls in table tennis is critical for ensuring fair play. This paper explores the development of an automated table tennis serve foul detection system leveraging 3D ball trajectory analysis and deep learning techniques. Using a multi-camera setup and a custom dataset, we employed YOLO for ball detection and Transformers for key trajectory point identification. The system achieved 87.52% precision in detecting fast-moving balls and an F1 score of 93 in recognizing critical serve points such as the throw, highest, and hit points. These results enable precise serve segmentation and robust foul detection based on criteria like toss height and vertical angle compliance. The approach simplifies traditional methods by focusing solely on the ball’s motion, eliminating computationally intensive pose estimation. Despite limitations such as a controlled experimental environment, the findings demonstrate the feasibility of AI-driven referee systems for table tennis, providing a foundation for broader applications in sports officiating.

**Keywords:** Table Tennis, Foul Detection, YOLO, 3D Trajectory Analysis, Multi-Camera System, Machine Learning, Transformer

1. Introduction

In competitive sports, ensuring fair play through consistent rule enforcement is essential for maintaining the integrity of the game. In table tennis, detecting serve fouls accurately is particularly challenging due to the high speed and complex dynamics of the ball and player movements [1]. Serve fouls, such as improper ball toss height, incorrect positioning, or a backward-angled toss, can provide players with unintended advantages. However, human judgment alone may struggle to detect these fouls accurately, especially in high-stakes or fast-paced matches. This has led to the need for automated foul detection systems that can bring precision, consistency, and impartiality to officiating in table tennis.

Accurate detection of serve fouls is vital to ensuring fair competition and compliance with table tennis regulations. Traditional manual methods for foul detection are prone to inconsistencies and errors due to the rapid movements involved in serves [2]. An automated system would provide a reliable, objective tool for detecting serve fouls, supporting referees in their decisions and enhancing the fairness of the game.

With advances in computer vision, new tools like YOLO11 (You Only Look Once, version 11) have shown significant improvements in real-time object detection. YOLO11, optimized for detecting small, fast-moving objects, offers the potential for precise and rapid ball tracking, making it particularly suitable for applications in sports. Additionally, multi-camera setups enable 3D reconstruction, capturing the full spatial dynamics of a table tennis serve from multiple perspectives. Despite these advancements, debates remain on optimal detection approaches, including whether single-camera or multi-camera systems provide the best accuracy and how well deep learning models can be adapted to the unique requirements of sports applications [3].

This study aims to create an automated table tennis serve foul detection system using 3D ball tracking, leveraging a multi-camera setup and deep learning techniques like YOLO11 for ball detection and Transformers for sequence analysis. By integrating YOLO11 with a multi-camera system, the proposed solution seeks to achieve high precision in detecting serve fouls by capturing the ball’s trajectory and player movements in 3D space.

Key contributions of this study include the development of a multi-camera-based 3D tracking system for table tennis and the application of YOLO11 for small object detection in fast motion. Additionally, the study explores using Transformer models to identify key points in the serve trajectory, enhancing the system’s ability to detect fouls based on movement patterns and ball positioning.

To guide our research, we introduce the following hypotheses:

* **Hypothesis 1**: can be further optimized to improve small object detection accuracy in high-speed ball tracking of table tennis.
* **Hypothesis 2**: Table tennis ball movement can be effectively used for video segmentation to isolate serve sequences.
* **Hypothesis 3**: Transformer models can accurately identify key turning points (throw, highest, and hit points) in the 3D trajectory of a table tennis serve.

2. Materials and Methods

2.1 Experimental Setup

A collage of a person standing in front of a table

Description automatically generated  
Figure 1. Multi-camera setup for table tennis serve foul detection. The cameras are positioned to capture different perspectives: two cameras are placed at the left and right sides of the table, while a third camera is mounted overhead.

The experimental setup consisted of three high-speed cameras, each connected via USB 3.0 to ensure minimal latency and capable of recording at 60 frames per second (fps). These cameras were strategically positioned around the table tennis table to capture the ball’s motion from multiple perspectives, as shown in Figure 1. The primary cameras, labeled as Camera 1 and Camera 2, were placed to the left and right of the table, focusing on the critical serve area. This placement enabled the capture of the ball’s trajectory from multiple angles and facilitated the calculation of 3D coordinates using triangulation, which was essential for accurate trajectory analysis and foul detection. The verification camera, labeled as Camera 3, was mounted on the ceiling above the athlete’s head, providing a top-down view that complemented the side views. While this overhead camera was not directly involved in 3D reconstruction, it improved tracking reliability by offering an additional perspective, especially useful in cases where the ball was occluded in the side views.

This multi-camera setup was chosen over a single-camera configuration to address limitations related to depth perception and occlusions. In high-speed sports like table tennis, achieving a precise 3D perspective of the ball’s trajectory and detecting fouls requires synchronized multi-camera views. The combination of these cameras ensured robust tracking across various angles, minimizing tracking loss and allowing accurate monitoring of the ball’s position, even in complex scenarios. To ensure a controlled experimental environment, the setup was conducted indoors with uniform lighting to minimize external interferences such as shadows and lighting variations. The table tennis table was placed at the center of the setup, with clear space around it to allow unobstructed views from all cameras.

Processing and analysis were conducted on a Windows 11 system equipped with a dedicated GPU (Graphics Processing Unit) to accelerate computationally intensive tasks such as real-time object detection and 3D reconstruction. Python and PyTorch served as the primary programming environment.

2.2 Data Collection and Synchronization

To accurately capture and synchronize the fast-paced movements involved in a table tennis serve, a careful setup for video recording and frame alignment was implemented. This process involved recording synchronized video feeds from multiple cameras and aligning frames across these feeds to enable precise 3D reconstruction of the ball’s trajectory[4].

OBS (Open Broadcaster Software) was employed to manage the multi-camera setup and ensure frame synchronization across all video feeds. By recording one composite video that includes views from all three cameras in designated sub-areas, OBS allowed for temporal alignment across all views (Figure 2). This composite video was later split into individual feeds for each camera, preserving synchronization.

A collage of a person using a computer

Description automatically generated

Figure 2. Frame alignment video recorded by OBS. The video layout shows the overhead view from the ceiling-mounted camera (left) and side views from the left and right cameras (right), allowing synchronized data capture across all views.

After recording, the composite video was split into individual video feeds for each camera while preserving the frame-by-frame synchronization established by OBS. This splitting process ensured that every frame from each camera was perfectly aligned in time, To confirm frame alignment, a visual inspection was performed across the individual video feeds. Key points in the serve sequence, such as the ball toss and hit moments, were compared across all views to ensure that these actions occurred simultaneously in each camera’s footage. This alignment is critical for accurate 3D trajectory reconstruction, Without precise alignment, any temporal mismatch between frames could lead to inaccuracies in calculating the ball’s 3D coordinates.

2.3 Calibration and 3D Reconstruction

Calibration involves two main components: intrinsic calibration to determine each camera’s internal parameters and extrinsic calibration to align the cameras with a common coordinate system.

Intrinsic calibration was performed using a chessboard pattern, which is a standard technique in computer vision for determining camera-specific parameters. The chessboard was placed within the view of each camera, and 100 images were captured. These images were then used to calculate each camera’s intrinsic parameters, including focal length, optical center, and lens distortion coefficients[5].

After obtaining intrinsic parameters, extrinsic calibration was conducted to establish a common 3D coordinate system across all cameras. This step involved identifying 16 fixed reference points around the table, as shown in Figure 3, and measuring their precise coordinates. These points were chosen based on their visibility across multiple camera views, ensuring they could serve as reliable references for spatial alignment. Each reference point's position was marked in the video, and its coordinates in the 3D space were recorded. By associating the 2D coordinates of these points in each camera view with their real-world 3D coordinates, the extrinsic parameters for each camera (rotation and translation vectors) were calculated.

A table tennis table with red squares

Description automatically generated

Figure 3. Extrinsic calibration points used for aligning cameras to a common coordinate system. The 16 fixed reference points are labeled with their 3D coordinates in the image, providing the foundation for accurate spatial calibration.

2.4 Ball Detection

YOLOv11 was selected for its high efficiency in detecting small, fast-moving objects, making it suitable for identifying a table tennis ball in each frame [6]. To optimize YOLOv11 for the specific challenges of this project, several modifications were implemented to improve its accuracy in detecting small objects like the table tennis ball, which is often difficult to track due to its rapid motion and small size in the frame.

The standard YOLOv11 architecture was adapted to enhance its sensitivity to small objects by removing the large object detection layers and incorporating a Resample Convolution (ResConv) layer, as shown in Figure 4. The ResConv layer is designed to better handle small-scale features, allowing the model to focus on the fine details required for detecting small objects like the ball. Additionally, Upsampling layers were modified to emphasize finer spatial resolution, which is crucial for capturing the ball's movement accurately, even at high speeds.

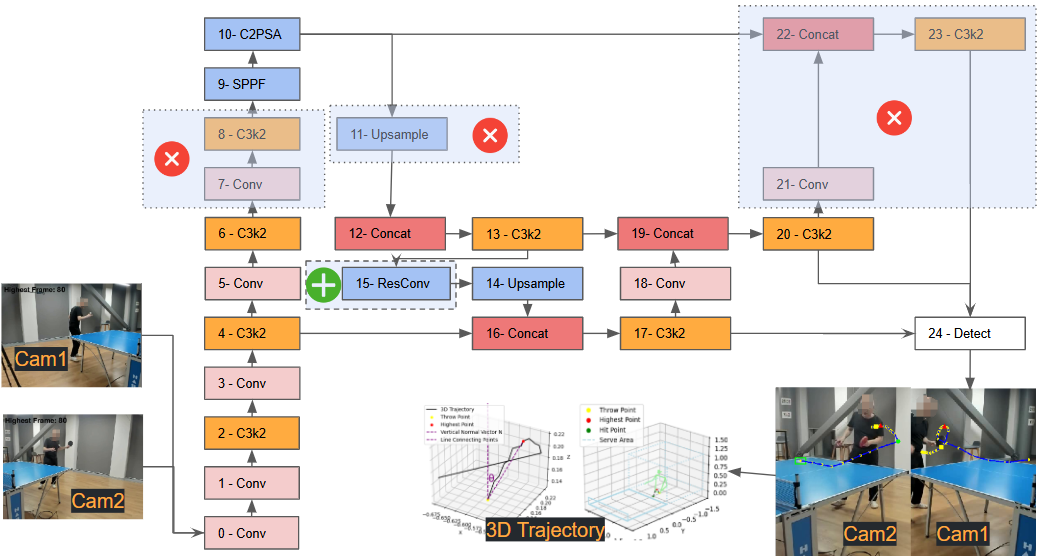


Figure 4. Modified YOLOv11 architecture for small object detection. The large object detection layer is removed, and a ResConv layer is added to enhance detection performance for small, fast-moving objects like the table tennis ball.

The ResConv layer is designed to enhance small object detection by preserving spatial details often lost during traditional downsampling. It divides the input tensor into smaller spatial components, retains localized features, and enriches the feature representation by concatenating these components along the channel dimension. A convolution operation then processes this enriched data to capture critical spatial relationships, improving the model’s ability to detect small objects.

(1)

Here, is the input tensor with shape (B,C,H,W), where B is the batch size, C the number of channels, and H,W the spatial dimensions. The input is sliced into four regions, capturing localized spatial information. These slices are concatenated along the channel dimension, resulting in a tensor of shape (B,4C,H/2,W/2), with quadrupled channels and halved spatial resolution.

The concatenated tensor is passed through a 3×3 convolution, producing the output Y with shape (B,ouc,H/2,W/2), where ouc is the number of output channels. This process preserves critical details, ensuring robust small object detection while maintaining computational efficiency.

With this optimized YOLOv11 model, the ball was detected simultaneously in frames from two different cameras (Cam1 and Cam2) positioned to capture the serve area from distinct angles. By detecting the ball from two synchronized perspectives, the system can leverage data from multiple views, ensuring consistent tracking and reducing the likelihood of detection loss, even during rapid ball movements.

After detecting the ball in both camera feeds, the system applied calibration theory to reconstruct the ball’s position in 3D space. Using the intrinsic and extrinsic calibration parameters obtained earlier, the 2D coordinates of the ball from each camera were mapped to a common 3D coordinate system. This triangulation process allowed for the precise calculation of the ball's position in 3D, enabling detailed trajectory analysis and accurate detection of key serve points, such as the throw, highest, and hit points.

2.5 Ball Tracking

ByteTrack was employed to maintain the table tennis ball’s position across consecutive frames [7]. ByteTrack is designed to associate every detection box, including low-confidence ones, improving its ability to handle objects with varying detection scores, this method has gained prominence for its robustness in tracking objects in challenging scenarios.

A diagram of a computer

Description automatically generated

Figure 5. ByteTrack Framework for Table Tennis Ball Tracking. The figure illustrates the workflow, starting from detection boxes output by YOLO. Detection boxes are categorized by confidence, with high-confidence boxes prioritized in the first association. Low-confidence detections are considered in a second association step, ensuring robust tracking. The Kalman filter maintains smooth track predictions, while unmatched tracks and detections are terminated or discarded.

ByteTrack operates in two stages, as illustrated in Figure 5. First, detection boxes from the YOLO detector are categorized based on their confidence scores into high-confidence and low-confidence groups. High-confidence detections are matched to existing tracks using IoU (Intersection over Union) criteria, ensuring that the most reliable detections are prioritized. The unmatched low-confidence boxes are then considered in a second round of association to address missed detections caused by occlusions, motion blur, or other factors. This two-stage matching process minimizes tracking loss and prevents objects from being prematurely discarded. The Kalman filter is integral to ByteTrack’s framework, predicting the positions of existing tracks to maintain consistency and smoothness across frames. Tracks that remain unmatched for extended periods are terminated, while unmatched detections that fail the second association are discarded to optimize performance.

Figure 5 provides an workflow overview of ByteTrack. Detection boxes from YOLO are categorized by detection scores into high-confidence and low-confidence groups. High-confidence detections undergo a first round of association with existing tracks, while low-confidence boxes are processed in a secondary matching step. The Kalman filter predicts track positions, and unmatched tracks or detections are discarded after multiple failed associations.

OpenCV's background subtraction and optical flow methods [8] provide a simpler alternative for motion detection but are susceptible to noise from lighting changes or background changes, which can cause track fragmentation. The experiment will compare it with ByteTrack.

2.6 Video Segmentation

the precise 3D trajectory of the table tennis ball, including its spatial coordinates and temporal sequence, is used as the primary cue for segmenting video. The trajectory spans the entire serve action, from the ball's appearance to its departure from the table. This approach significantly simplifies video segmentation in the context of table tennis serves, eliminating the need for multi-stream models commonly employed in action segmentation studies.

Unlike recent methods that heavily rely on player pose estimation and multi-stream architectures integrating RGB data, optical flow, and player positioning (as seen in table tennis and tennis research) [9-12], this study focuses exclusively on the ball’s 3D trajectory. The calibrated 3D coordinates of the ball provide sufficient information to isolate serve sequences without relying on additional data streams or complex model architectures.

The segmentation process is straightforward and relies on the ball’s spatial movement within the calibrated 3D space. A threshold of Y>0.5, representing a spatial boundary in the serve area, is used to identify the serve sequence. This criterion isolates the serve action by focusing on the ball’s trajectory during its active involvement in the serve, effectively capturing the temporal and spatial characteristics of the sequence.

2.7 Transformer Model for Key Point Detection

In this study, a Transformer model was applied to analyse the 3D trajectory of a table tennis serve and identify key turning points, such as the throw, highest, and hit points. This approach leverages the Transformer’s attention mechanism[13], which excels in capturing dependencies in sequential data, making it highly effective for recognizing patterns in the ball’s movement over time. By learning from labeled trajectories, the model can accurately predict critical points within the serve sequence, thereby enhancing the system’s ability to detect fouls based on serve dynamics.

A diagram of a flowchart

Description automatically generated

Figure 6. Illustration of the Transformer's role in identifying turning points in the ball's 3D trajectory. The left panel shows the trajectory path with spatial axes (X, Y, Z), while the right panel highlights the annotated turning points (throw point, highest point, and hit point). The schematic of the Transformer model outlines its components, such as multi-head attention, which enables it to process the trajectory data and extract key turning points.

As shown in Figure 6, the Transformer architecture is visualized alongside the 3D trajectory. The trajectory highlights the critical turning points: the throw point (marked as a yellow square), the highest point (marked as a red triangle), and the hit point (marked as a blue circle). The left panel depicts the 3D path of the ball, while the right panel overlays the annotated turning points on the trajectory. The Transformer framework on the right demonstrates how the positional encoding and multi-head attention layers enable the model to process sequential data and extract these key moments with precision.

Studies like S2TNet and the Trajectory Unified Transformer [14-16] apply Transformer-based models to predict future positions in human motion or traffic data by identifying significant turning points. While these applications involve complex social interactions or environmental factors, the focus here is on the ball’s motion alone, allowing the Transformer to concentrate on the trajectory patterns specific to the serve.

The Transformer model in this study was trained on a dataset of labeled trajectories, where each serve sequence was annotated with the critical points of interest: the throw point (where the ball begins its upward toss), the highest point (the peak of the ball's toss), and the hit point (where the player makes contact). These labels enabled the model to learn the specific patterns associated with each turning point in the serve. The attention mechanism of the Transformer allowed it to discern subtle changes in the ball’s movement over time, capturing dependencies that traditional models might overlook. This ability to track complex temporal dependencies made the Transformer model particularly well-suited for accurately identifying turning points in the trajectory.

2.8 Rule-Based Foul Detection

The system employs a set of rule-based criteria to determine whether a serve violates table tennis regulations. These criteria are designed to assess various aspects of the serve and provide objective data to assist referees in evaluating serve compliance[17,18].

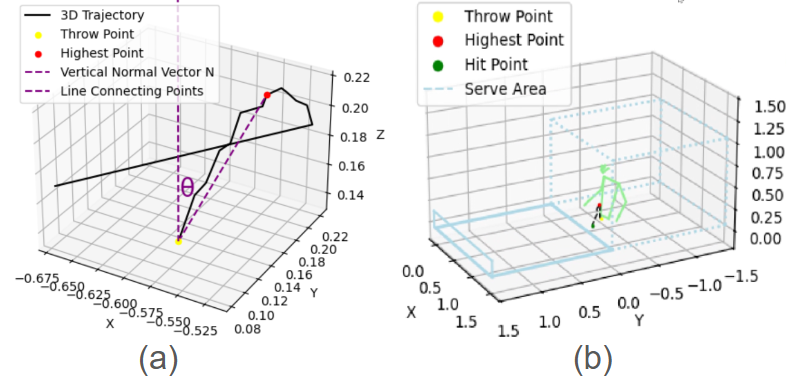


Figure 6. Rule-Based Foul Detection System:(a) 3D trajectory visualization with the throw point (yellow), highest point (red), and the calculated vertical angle . The vertical normal vector and the line connecting the key points are included for clear representation. (b) Spatial analysis of the serve area, showing the trajectory of the ball relative to the permitted boundaries and confirming compliance with service area rules.

1. Minimum Drop Height: To ensure that the serve meets the minimum height requirement, the system monitors the ball trajectory to confirm it reaches a specified drop height. This criterion helps verify that the serve complies with the rules regarding the initial toss height.
2. Vertical Angle at Throw: which assesses the initial toss direction. To ensure that the toss remains near the vertical axis, the system calculates the angle between the vector connecting the throw point and the highest point of the ball’s trajectory, and the vertical reference vector (0,0,1). This angle is computed using the following formula:

(1)

Here, , , , ​ are the components of the vector ,, which connects the throw point to the highest point, and , , represent the components of the vertical reference vector = (0, 0, 1). This formula computes the angle in radians, which is then converted to degrees. If exceeds 30°, the toss is flagged as a potential foul for excessive backward tilt. As illustrated in Figure 7(a), the system visualizes the ball’s 3D trajectory, indicating key points such as the throw point (yellow) and highest point (red), along with the calculated vertical angle and the vertical reference vector. This visualization aids in understanding and validating the system’s assessment of the toss angle.

1. Service Area Positioning: The 3D coordinates of the ball are continuously tracked to ensure that the ball remains within the designated service area throughout the serve. as shown in Figure 7(b). This spatial analysis prevents serves from originating outside the legal area, ensuring compliance with spatial regulations.

3. Results

In this section, we provide a detailed analysis of the results produced by the models we trained. All training results refer to the model being trained on our dataset for 300 epochs, with a batch size of 16.

**3.1 Ball Detection**

The performance of our table tennis ball detection system was evaluated by comparing it with two benchmark models: YOLOv8 and YOLO11m. The evaluation focused on metrics such as precision, recall, mAP@50, mAP@50:95, and training time. As shown in **Table 1**, our model achieved the highest detection accuracy across all metrics, demonstrating its superiority in handling the fast-moving, small-sized table tennis ball while maintaining competitive training efficiency. Specifically, our model achieved a precision of 87.52%, a recall of 83.37%, an mAP@50 of 86.87%, and an mAP@50:95 of 39.84%, outperforming both YOLOv8 and YOLO11m. The training time of our model was 4 hours and 33 minutes, slightly longer than YOLOv8 but shorter than YOLO11m, reflecting a balance between accuracy and efficiency.

**Table 1.** Performance values of YOLOv8, YOLO11m and Ours

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Precision** |  | **Recall** | **mAP50** | **mAP50-95** | **Training Time** |
| **YOLOv8** | 84.92% |  | 75.40% | 81.30% | 36.30% | 3hrs24mins |
| **YOLO11m** | 81.25% |  | 81.25% | 84.12% | 38.12% | 4hrs59mins |
| **Ours** | 87.52% |  | 83.37% | 86.87% | 39.84% | 4hrs33mins |

The progression of these metrics over 300 training epochs is illustrated in Figure 7, which provides a detailed comparison of the models. The precision plot shows that our model maintained consistently higher precision throughout the training process, indicating its ability to minimize false positives in detecting table tennis balls. Similarly, the recall plot demonstrates the capability of our model to detect a larger proportion of true positives compared to the other models. In the mAP@50 plot, our model exhibits superior accuracy with less fluctuation, highlighting its robustness in detecting objects with an IoU threshold of 0.5. For the stricter mAP@50:95 metric, our model shows steady improvement, surpassing the performance of YOLOv8 and YOLO11m, and confirming its reliability across a range of IoU thresholds.

A group of graphs with different colored lines

Description automatically generated

**Figure 7. Performance Metrics Over Training Epochs for YOLOv8, YOLO11m, and Our Model.** This figure illustrates the progression of key metrics (precision, recall, mAP@50, and mAP@50:95) over 300 training epochs. Our model outperforms YOLOv8 and YOLO11m across all metrics, demonstrating its robustness and accuracy in table tennis ball detection.

The comparative analysis reveals that while YOLOv8 benefits from a shorter training time, it sacrifices detection accuracy, particularly in recall and mAP metrics. YOLO11m, on the other hand, achieves improved recall and mAP but at the expense of a longer training time. In contrast, our model strikes an optimal balance, delivering superior detection performance while maintaining reasonable training time. The enhanced results of our model can be attributed to its optimized detection layers and robust association mechanisms, which are particularly effective in tracking small, fast-moving objects like table tennis balls.

**3.4. Ball Tracking**

The effectiveness of ball tracking methods in high-speed table tennis scenarios was evaluated using two approaches: OpenCV background subtraction with optical flow and Byte-Track. Each method's performance is illustrated in Figure 8, which provides a side-by-side comparison of their tracking capabilities. The left panel demonstrates the results of background subtraction with optical flow, while the right panel highlights the superior performance of Byte-Track.

A collage of a person playing ping pong

Description automatically generated

Figure 8. Comparison of Ball Tracking Methods:  
(left) OpenCV background subtraction and optical flow method resulting in fragmented and inconsistent ball trajectories due to background noise, rapid motion, and interference from moving players.  
(right) Byte-Track method delivering smoother and continuous ball trajectories, robustly handling complex motion and maintaining tracking consistency.

Background subtraction and optical flow, implemented using OpenCV, is a relatively straightforward technique for detecting moving objects in a static or semi-static environment. This method isolates areas of motion by identifying changes between consecutive frames, allowing for the detection of objects like a table tennis ball. In this study, it effectively captured the ball's movement against a stable background, as shown in the left panel of Figure 8. However, this approach exhibited several limitations. It was highly sensitive to background noise, such as shadows and reflections, which often led to fragmented detection results. Additionally, the presence of moving players in the background introduced significant interference, making it challenging for the system to distinguish between the ball and player movements. This interference frequently disrupted the tracking process, causing the trajectory to break or become inconsistent. Combined with the rapid and complex motion of the ball during a serve, these issues severely impacted the continuity of the tracking, hindering precise trajectory analysis.

In contrast, Byte-Track demonstrated superior performance by providing smoother and more continuous ball tracking, as depicted in the right panel of Figure 8. Byte-Track employs an advanced tracking mechanism that associates detected objects across frames using features such as object size, movement consistency, and positional prediction. This robust framework ensures track continuity, even during rapid directional changes or complex motion patterns. Unlike background subtraction, Byte-Track effectively handles the high-speed dynamics of a table tennis serve, maintaining a stable and uninterrupted trajectory. Moreover, Byte-Track’s ability to focus on the ball as the primary object of interest mitigates the impact of background player movement, ensuring reliable tracking.

**3.5. Video Segmentation Results**

A person playing ping pong

Description automatically generated

Figure 8. Video Segmentation Results:  
Top: Synchronized left and right camera views showing key trajectory points during a table tennis serve. The throw point (yellow), highest point (red), and hit point (green) are marked to track the ball’s trajectory.  
Bottom: Segmented timeline showing frame classification as "No Foul" (gray) and "Foul" (blue), with transitions aligned to the serve events.

The video segmentation results demonstrate the effectiveness of the proposed system in identifying key events during a table tennis serve and distinguishing between fouls and compliant serves. The segmentation process involves analyzing the 3D trajectory of the ball and detecting critical points, such as the throw point, the highest point, and the hit point, as shown in Figure 9. Additionally, the system classifies frames as either "No Foul" or "Foul," providing a detailed visual representation of the serve sequence.

In Figure 9, the left and right camera views are synchronized to provide a comprehensive analysis of the serve. Key trajectory points are highlighted, with the yellow marker representing the throw point, the red marker indicating the highest point, and the green marker showing the hit point. This multi-camera approach ensures that all relevant motion events are accurately captured and analyzed. The segmented timeline below the video frames provides a clear visualization of the classification results. Blue segments indicate fouls, while gray segments represent no-foul frames. The system demonstrates its ability to identify transitions between compliant and non-compliant actions seamlessly.

The segmentation is further validated by the system's robust identification of critical trajectory points, ensuring that serve sequences are broken down into precise segments. By leveraging the ball’s trajectory and synchronized video data, the system achieves reliable segmentation even in scenarios with rapid motion or occlusions. This segmentation capability is essential for applications like foul detection and performance analysis, where detailed frame-level analysis is required.

**3.6. Transformer Find Turning Points**

The results, as shown in Figures 11 and 12, demonstrate the effectiveness of the Transformer model in identifying key turning points during table tennis serves through synchronized multi-camera views, 3D trajectory analysis, and detailed serve statistics.

After the Transformer model calculated the turning points of the 3D trajectory, the corresponding frames were retrieved from two cameras (Cam1 and Cam2) based on the frame indices of the throw point, highest point, and hit point. These frames are displayed in Figure 11, showing synchronized views for the detected turning points. Manual verification of these turning points yielded an F1 score of 93, confirming the accuracy of the model’s predictions and its reliability in detecting key moments during the serve.

Figure 12 provides a more comprehensive analysis of the turning points within the context of the 3D trajectory and foul detection. The central 3D trajectory plot visually illustrates the ball’s motion, with key turning points (throw, highest, and hit points) highlighted. The plot also shows their spatial alignment with the serve area boundary, depicted as a dotted box. This boundary represents the permitted serve area, allowing the system to validate compliance with spatial rules. For example, the throw point and hit point fall within the boundary, ensuring the serve adheres to spatial constraints.

The final results of the experiment, as shown in Figure 12, summarize all serve statistics during the athlete’s training session. The timeline clearly shows transitions between compliant frames (No Foul) and non-compliant frames (Foul), providing a visual representation of rule compliance. Multiple fouls were detected during the training session, along with the number of times each foul triggered its respective rule. The current serve action is also analyzed in detail, showing the 3D trajectory, the 3D coordinates of the three key turning points, the tossed upward distance, and the angle with the vertical axis.

A collage of a person playing ping pong

Description automatically generated

Figure 11. Multi-Camera Frames of Turning Points:  
Frames from Cam1 and Cam2 corresponding to the throw point (yellow), highest point (red), and hit point (green), as detected by the Transformer model. The high F1 score (93) validates the accuracy of these turning points.

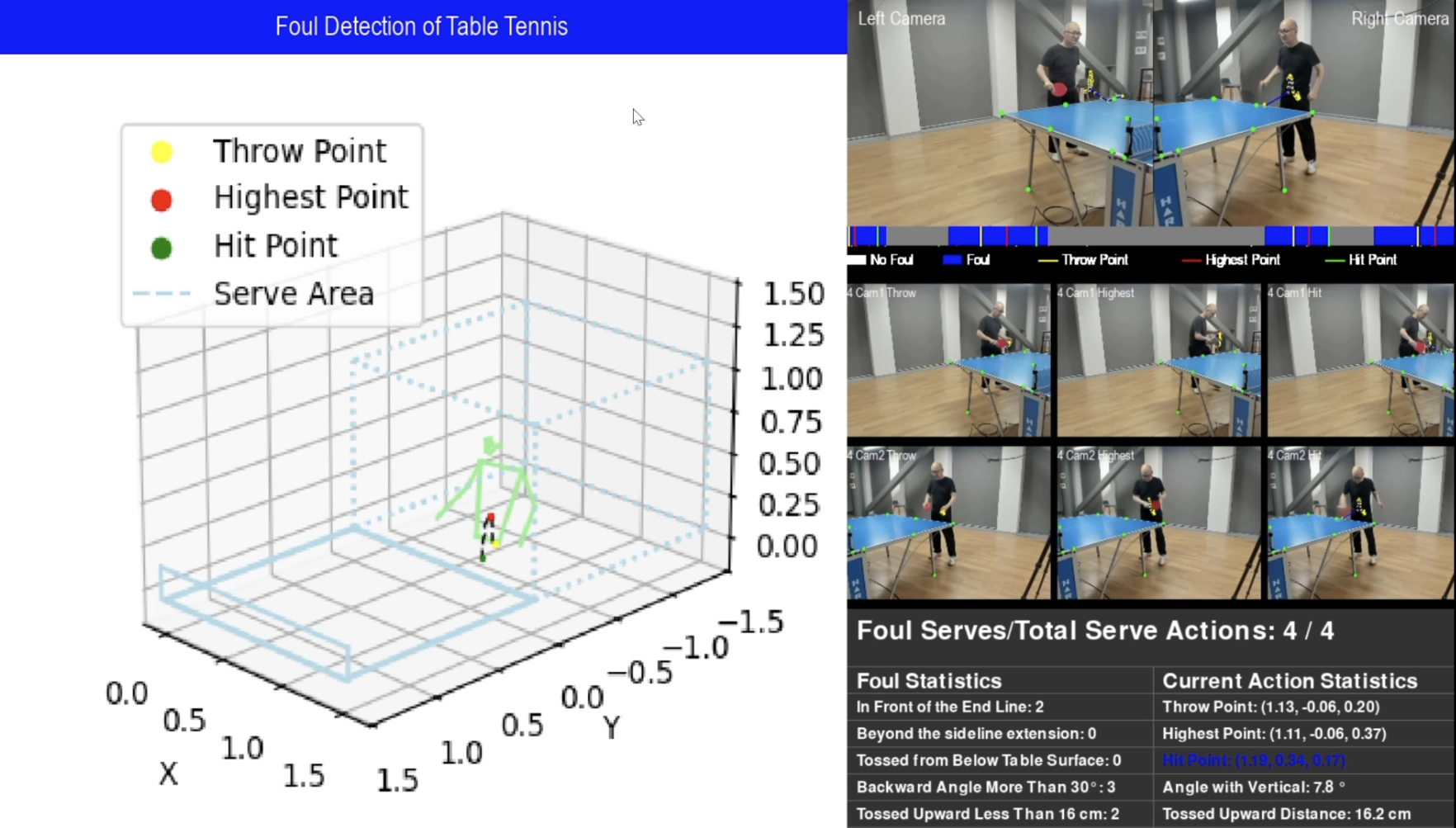


Figure 12. 3D Trajectory and Serve Statistics:  
The central 3D trajectory plot highlights the ball's motion and key turning points, aligned with the spatial limits of the serve area (dotted boundary box). The right panel provides serve statistics, showing the timeline of compliant and non-compliant frames, foul counts, and detailed metrics for the current serve, including tossed upward distance and the angle with the vertical axis.

4. Discussion

This project investigates the application of deep learning models for estimating the trajectory and detecting fouls during table tennis serves. To achieve this, a unique dataset was generated using self-recorded videos captured by a multi-camera setup, specifically designed for high-speed motion tracking. The dataset was used to train various deep learning models, including YOLO for detection and Transformers for key turning point analysis, and the trained models were tested in a real-time environment. In this discussion, we analyze the results, reflect on the project’s limitations, and propose future directions, focusing on the three hypotheses outlined earlier.

4.1 Can be further optimized to improve small object detection accuracy in specific scenarios, such as high-speed ball tracking in table tennis?

The detection and tracking of small, fast-moving objects like a table tennis ball pose unique challenges, particularly in high-speed scenarios. Our model demonstrated significant advancements, achieving a precision of 87.52% and recall of 83.37%, outperforming YOLOv8 and YOLO11m in mAP@50 (86.87%) and mAP@50:95 (39.84%) while maintaining a reasonable training time of 4 hours and 33 minutes. Experimental results confirmed that our enhancements for small object detection significantly improved accuracy, and the removal of large object detection layers also reduced training time. These optimizations suggest that YOLO11 can be further tailored for different usage scenarios.

4.2 Table tennis ball movement can be effectively used for video segmentation to isolate serve sequences?

Leveraging the 3D trajectory of the ball, this study avoided the complexity of multi-stream networks typically used for video segmentation, which often rely on RGB, optical flow, and player pose data. However, such methods come with significant computational costs and require complex architectures. In contrast, our approach simplifies segmentation by focusing solely on the ball's motion. This method capitalizes on the distinct and consistent movement patterns of the ball during serves, enabling accurate isolation of serve sequences without requiring player pose data or large training datasets. The streamlined methodology not only maintains high segmentation accuracy but also reduces computational demands, making it well-suited for real-time applications in table tennis.

4.3 Transformer models can accurately identify key turning points (throw, highest, and hit points) in the 3D trajectory of a table tennis serve.

The Transformer model performed exceptionally well, achieving an F1 score of 93 in manual validation of the detected turning points. The use of self-attention mechanisms allowed the model to capture temporal dependencies in the trajectory data effectively, resulting in precise identification of these critical moments. This capability is essential for applications like foul detection and serve analysis, where turning points play a pivotal role in determining compliance with game regulations. However, while the results are promising, additional experiments with larger and more diverse datasets are necessary to validate the model’s generalizability across different playing conditions and ball trajectories.

4.5 Limitations and future work

Despite the promising results, the project has certain limitations that must be acknowledged. The primary challenge lies in the limited size and diversity of the dataset, which was collected in a controlled indoor environment. This restricts the model's ability to generalize to outdoor settings or matches involving different lighting conditions.

Future work will focus on reducing the complexity of the environment setting so that the system can adapt to different table tennis court conditions. In addition, expanding more foul detection rules will improve the versatility of the model.

5. Conclusion

In this study, we developed a table tennis serve foul detection system that leverages 3D trajectory analysis and deep learning models, including YOLO for ball detection and Transformers for critical turning point identification. The system achieved significant milestones: YOLO-based detection achieved 87.52% accuracy and 83.37% recall, demonstrating high accuracy in tracking small and fast-moving balls. The Transformer model further achieved an F1 score of 93 when detecting turning points, such as the throw point, the highest point, and the impact point. The system effectively segments the serve sequence using only the ball's trajectory, eliminating the need for computationally expensive pose estimation, while providing robust rule compliance analysis. The experiment accurately identified the number of fouls and accurately linked the serve action to the rule violation, including the toss height and back tilt angle. Despite limitations, such as the controlled environment of the dataset, the system demonstrates the feasibility of AI as a table tennis referee.

**Author Contributions:** Conceptualization, G.L.Y. and W.Q.Y.; methodology, G.L.Y.; software, G.L.Y.; validation, G.L.Y.; formal analysis, G.L.Y.; investigation, G.L.Y.; resources, G.L.Y., W.Q.Y.; data collection, G.L.Y.; writing—original draft preparation, G.L.Y.; writing—review and editing, G.L.Y. W.Q.Y.; visualization, G.L.Y.; supervision, W.Q.Y.; project administration W.Q.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has no external fundings.

**Data Availability Statement:** Data sharing is not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

References

1. Zhou, H.; Nguyen, M.; Yan, W. Q. Computational Analysis of Table Tennis Matches from Real-Time Videos Using Deep Learning. In *Image and Video Technology*; Yan, W. Q., Nguyen, M., Nand, P., Li, X., Eds.; Springer Nature: Singapore, 2024; pp 69–81. <https://doi.org/10.1007/978-981-97-0376-0_6>.
2. Hung, C.-H. A Study of Automatic and Real-Time Table Tennis Fault Serve Detection System. *Sports* **2018**, *6* (4), 158. <https://doi.org/10.3390/sports6040158>.
3. Yan, W. Q. *Computational Methods for Deep Learning: Theory, Algorithms, and Implementations*; Springer Nature, 2023; pp. 154–196.
4. Poliakov, A.; Marraud, D.; Reithler, L.; Chatain, C. Physics Based 3D Ball Tracking for Tennis Videos. In *2010 International Workshop on Content Based Multimedia Indexing (CBMI)*; 2010; pp 1–6. <https://doi.org/10.1109/CBMI.2010.5529897>.
5. Zhang, Y.-J. Camera Calibration. In *3-D Computer Vision: Principles, Algorithms and Applications*; Zhang, Y.-J., Ed.; Springer Nature: Singapore, 2023; pp 37–65. <https://doi.org/10.1007/978-981-19-7580-6_2>.
6. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo Algorithm Developments. *Procedia Computer Science* **2022**, *199*, 1066–1073. <https://doi.org/10.1016/j.procs.2022.01.135>.
7. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. In *Computer Vision – ECCV 2022*; Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., Hassner, T., Eds.; Springer Nature Switzerland: Cham, 2022; pp 1–21. <https://doi.org/10.1007/978-3-031-20047-2_1>.
8. Barron, J. L.; Fleet, D. J.; Beauchemin, S. S. Performance of Optical Flow Techniques. *Int J Comput Vision* **1994**, *12* (1), 43–77. <https://doi.org/10.1007/BF01420984>.
9. Ding, G.; Sener, F.; Yao, A. Temporal Action Segmentation: An Analysis of Modern Techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2024**, *46* (2), 1011–1030. <https://doi.org/10.1109/TPAMI.2023.3327284>.
10. Bian, J.; Li, X.; Wang, T.; Wang, Q.; Huang, J.; Liu, C.; Zhao, J.; Lu, F.; Dou, D.; Xiong, H. P2ANet: A Large-Scale Benchmark for Dense Action Detection from Table Tennis Match Broadcasting Videos. *ACM Trans. Multimedia Comput. Commun. Appl.* **2024**, *20* (4), 118:1-118:23. <https://doi.org/10.1145/3633516>.
11. Tran, T.-D. TTNet: A Novel Machine Learning Model for Facial Emotion Detection in Online Learning Systems. *SoftwareX* **2024**, *27*, 101787. <https://doi.org/10.1016/j.softx.2024.101787>.
12. Martin, P.-E.; Benois-Pineau, J.; Péteri, R.; Morlier, J. Three-Stream 3D/1D CNN for Fine-Grained Action Classification and Segmentation in Table Tennis. In *Proceedings of the 4th International Workshop on Multimedia Content Analysis in Sports*; MMSports’21; Association for Computing Machinery: New York, NY, USA, 2021; pp 35–41. <https://doi.org/10.1145/3475722.3482793>.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł. ukasz; Polosukhin, I. 00-Attention Is All You Need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2017; Vol. 30.
14. Hu, S.; Shen, L.; Zhang, Y.; Chen, Y.; Tao, D. On Transforming Reinforcement Learning With Transformers: The Development Trajectory. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2024**, *46* (12), 8580–8599. <https://doi.org/10.1109/TPAMI.2024.3408271>.
15. Chen, W.; Wang, F.; Sun, H. S2TNet: Spatio-Temporal Transformer Networks for Trajectory Prediction in Autonomous Driving. In *Proceedings of The 13th Asian Conference on Machine Learning*; PMLR, 2021; pp 454–469.
16. Shi, L.; Wang, L.; Zhou, S.; Hua, G. Trajectory Unified Transformer for Pedestrian Trajectory Prediction; In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* , 2023, pp. 9675-9684.
17. *Document | International Table Tennis Federation*. <https://documents.ittf.sport/document/341> (accessed 2024-11-18).
18. Nasution, U.; Nasution, M. A. H.; Habibi, M. I.; Tahira, W. L. A.; Ridoh, M. Analysis of the Development of Regulations and Policies in the World of Table Tennis: A Literature Study Approach. *Journal Coaching Education Sports* **2024**, *5* (1), 25–32. <https://doi.org/10.31599/47rkgk16>.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.