electronics

MDPI

Article

Foul Detection for Table Tennis Serves Using Deep Learning

Guang Liang Yang, Minh Nguyen, Wei Qi Yan * and Xue Jun Li *

School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand; fdp5284@autuni.ac.nz (G.L.Y.); minh.nguyen@aut.ac.nz (M.N.)
* Correspondence: weiqi.yan@aut.ac.nz (W.Q.Y.); xuejun.li@aut.ac.nz (X.J.L.)

Abstract: Detecting serve fouls in table tennis is critical for ensuring fair play. This paper explores the development of foul detection of table tennis serves by leveraging 3D ball trajectory analysis and deep learning techniques. Using a multi-camera setup and a custom dataset, we employed You Only Look Once (YOLO) models for ball detection and Transformers for critical trajectory point identification. We achieved 87.52% precision in detecting fast-moving balls and an F1 score of 0.93 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 motion, eliminating computationally intensive pose estimation. Despite limitations such as a controlled experimental environment, the findings demonstrate the feasibility of artificial intelligence (AI)-driven referee systems for table tennis games, 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 sports games, 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.

In recent years, significant progress has been made in the analysis of sports using computer vision and deep learning techniques. In table tennis, studies such as Kulkarni et al. have focused on stroke detection and recognition, leveraging ball trajectory data with advanced models like YOLOv4 and TrackNetv2 [2]. Similarly, Voeikov et al. proposed TTNet, a real-time neural network for temporal and spatial video analysis, designed for event spotting and ball tracking in table tennis matches. In addition to table tennis, ball-tracking methods in other sports provide valuable insights [3]. For instance, Huang et al. developed TrackNet, a deep learning model for high-speed and small object tracking in tennis, achieving high precision with minimal equipment [4]. Caio et al. demonstrated a context-aware deep learning approach for 3D localization of basketballs using calibrated images, highlighting the potential for accurate object tracking in fast-paced scenarios [5].

Academic Editor: Ping-Feng Pai

Received: 20 November 2024
Revised: 14 December 2024
Accepted: 24 December 2024
Published: 25 December 2024

Citation: Yang, G.L.; Nguyen, M.; Yan, W.Q.; Li, X.J. Foul Detection for Table Tennis Serves Using Deep Learning. *Electronics* **2025**, *14*, 27. https://doi.org/10.3390/electronics14010027

Despite these advances, most methods focus on general tracking and event spotting, with limited attention to serve foul detection. This study aims to address this gap by combining 3D trajectory analysis and deep learning techniques for automated foul detection in table tennis, offering a novel and specialized solution.

Accurate detection of serve fouls is vital to ensuring fair competition and compliance with table tennis regulations. Traditional methods for foul detection are prone to inconsistencies and errors due to the rapid movements involved in serves [6]. 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 You Only Look Once (YOLO), 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, a multi-camera setup enables 3D reconstruction, capturing the full spatial dynamics of table tennis serves 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 [7].

This paper aims to detect serve fouls for table tennis games using three-dimensional (3D) ball tracking. It leverages a multi-camera setup and deep learning methods 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 trajectory of ball and player movements in 3D space.

Key contributions of this paper include developing a multi-camera-based 3D tracking for table tennis balls and applying YOLO11 for small object detection in fast motion. Additionally, we explore and identify key points by using Transformer models in the serve trajectory, enhancing the ability to detect fouls based on movement patterns and ball positioning.

2. Materials and Methods

2.1. Experimental Setup

The experimental setup consists of three high-speed USB cameras: two Logitech Brio 4K cameras, each operating at 90 frames per second (fps), and one Razer Kiyo Pro Ultra camera, operating at 60 fps. These frame rates were selected to ensure the temporal resolution needed to capture the rapid movements of the table tennis ball during high-speed serves and rallies. Each camera was connected via USB 3.0 to ensure minimal latency.

The cameras were strategically positioned around the table tennis table to capture the motion of the ball from multiple perspectives, as shown in Figure 1. The primary cameras, labeled Camera 1 and Camera 2, were placed to the left and right of the table, focusing on the critical serve area. This arrangement enabled the capture of the trajectory of the ball 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 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, particularly 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 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.

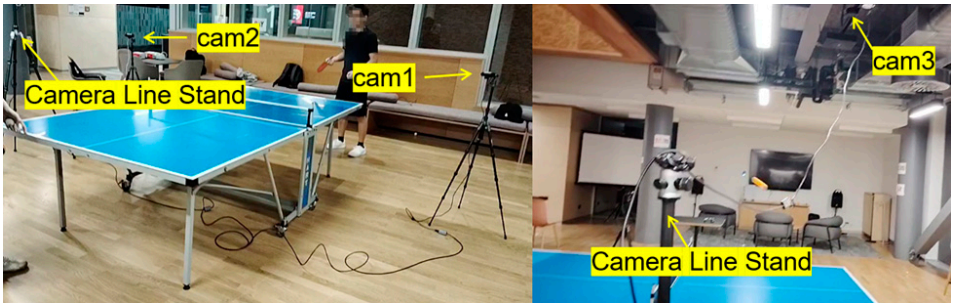


Figure 1. Multi-camera setup for table tennis serve foul detection. The cameras are positioned to capture different perspectives: Two cameras are placed at the left and right sides of the table, while a third camera is mounted overhead.

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 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 an NVIDIA GeForce RTX 4060 Laptop GPU 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

A setup for video recording and frame alignment was implemented to capture and synchronize the fast-paced movements involved in a serve of table tennis. This process involved recording synchronized video feeds from multiple cameras and aligning frames across these feeds to enable precise 3D reconstruction of the ball trajectory [8].

To achieve precise synchronization of video streams, we used the moment of ball contact with the table, captured by two cameras, as the alignment reference point. This ensured that the frames from both video streams were perfectly aligned without any temporal misalignment. With this synchronization approach, there was no displacement error caused by time shifts between the cameras, ensuring the accuracy of 3D trajectory reconstruction.

Open Broadcaster Software (OBS) is used 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.

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. The 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. This alignment is critical for accurate 3D trajectory reconstruction; without precise alignment, any temporal mismatch between frames could lead to inaccuracies in calculating the 3D coordinates of the ball.

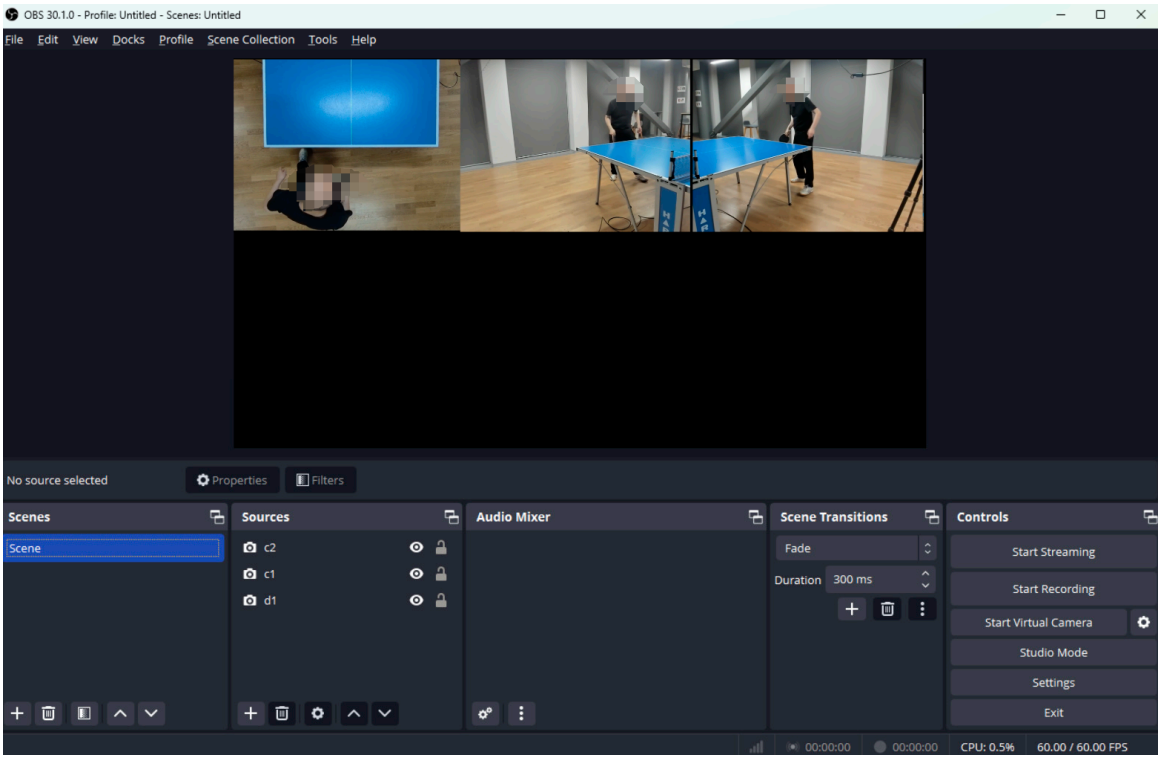


Figure 2. Frame alignment video recorded by OBS. The video layout shows the overhead view from the ceiling-mounted camera (**left**) and side views from the left and right cameras (**right**), allowing synchronized data capture across all views.

This dual synchronization approach—using OBS for initial alignment and ball contact for final validation—was critical for accurate 3D trajectory reconstruction; without precise alignment, any temporal mismatch between frames could lead to inaccuracies in calculating the 3D coordinates of the ball.

2.3. Calibration and 3D Reconstruction

Calibration involves two main components: intrinsic calibration, which determines the internal parameters of each camera, and extrinsic calibration, which aligns the cameras with a common coordinate system.

Intrinsic calibration was performed by using a chessboard pattern, a standard computer-vision technique for determining camera-specific parameters. The chessboard was placed within the view of each camera, and 100 images were captured. These images were then employed to calculate each camera's intrinsic parameters, including focal length, optical center, and lens distortion coefficients [9].

After obtaining intrinsic parameters, extrinsic calibration was conducted to establish a standard 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 with the real-world 3D coordinates, the extrinsic parameters for each camera (rotation and translation vectors) were calculated.

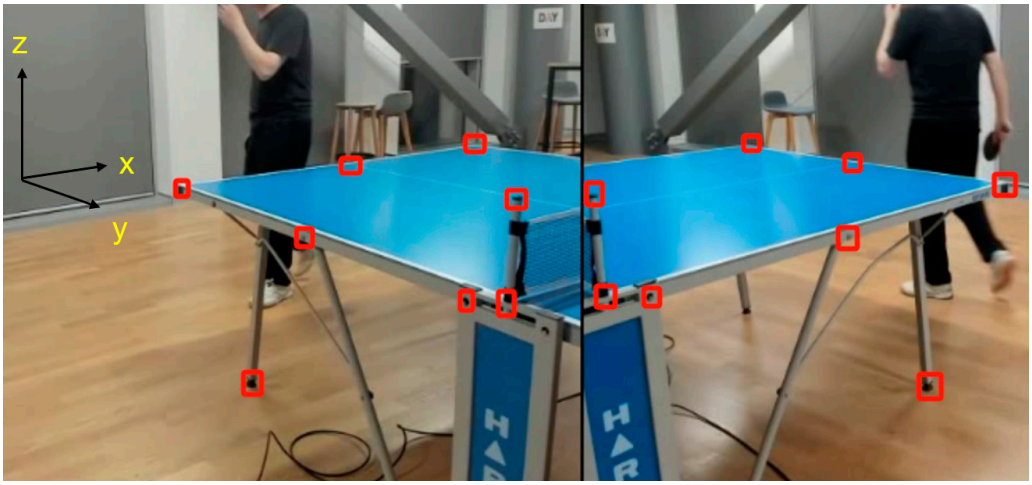


Figure 3. The calibration points are applied to align cameras with a coordinate system. The 16 fixed reference points are labeled with their 3D coordinates in the image, providing the foundation for accurate spatial calibration.

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 [10]. 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.

To adapt the YOLOv11 network for our specific task, we performed transfer learning using a custom dataset of 2000 labelled images extracted from the experimental videos. The images were resized to 640 $\times$ 640 pixels, the standard input resolution for YOLO, and selected to represent a wide variety of ball movements, including serves, smashes, and rallies. Each image was manually annotated with bounding boxes for the ball. The dataset was divided into 70% for training, 20% for validation, and 10% for testing. Training was conducted for 300 epochs with a batch size of 16, using a cosine annealing learning rate schedule starting at 0.001. To ensure efficient processing, training was performed on an NVIDIA A100 GPU, with loss and accuracy metrics monitored during training and validation to prevent overfitting.

The standard YOLOv11 architecture was adapted to enhance its sensitivity to small objects by removing the large object detection layers and incorporating a custom 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 movement of the ball accurately, even at high speeds.

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 and improve the model's ability to detect small objects.

$$Y = Conv(concat(X[\ldots, :: 2, :: 2], X[\ldots, 1 :: 2, :: 2], X[\ldots, :: 2, 1 :: 2], X[\ldots, 1 :: 2, 1 :: 2])) \quad (1)$$

where X is the input tensor with shape (B, C, H, W), B is the batch size, C is the number of channels, and H are W are 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 x' of shape $(B, 4C, H/2, W/2)$, with quadrupled channels and halved spatial resolution.

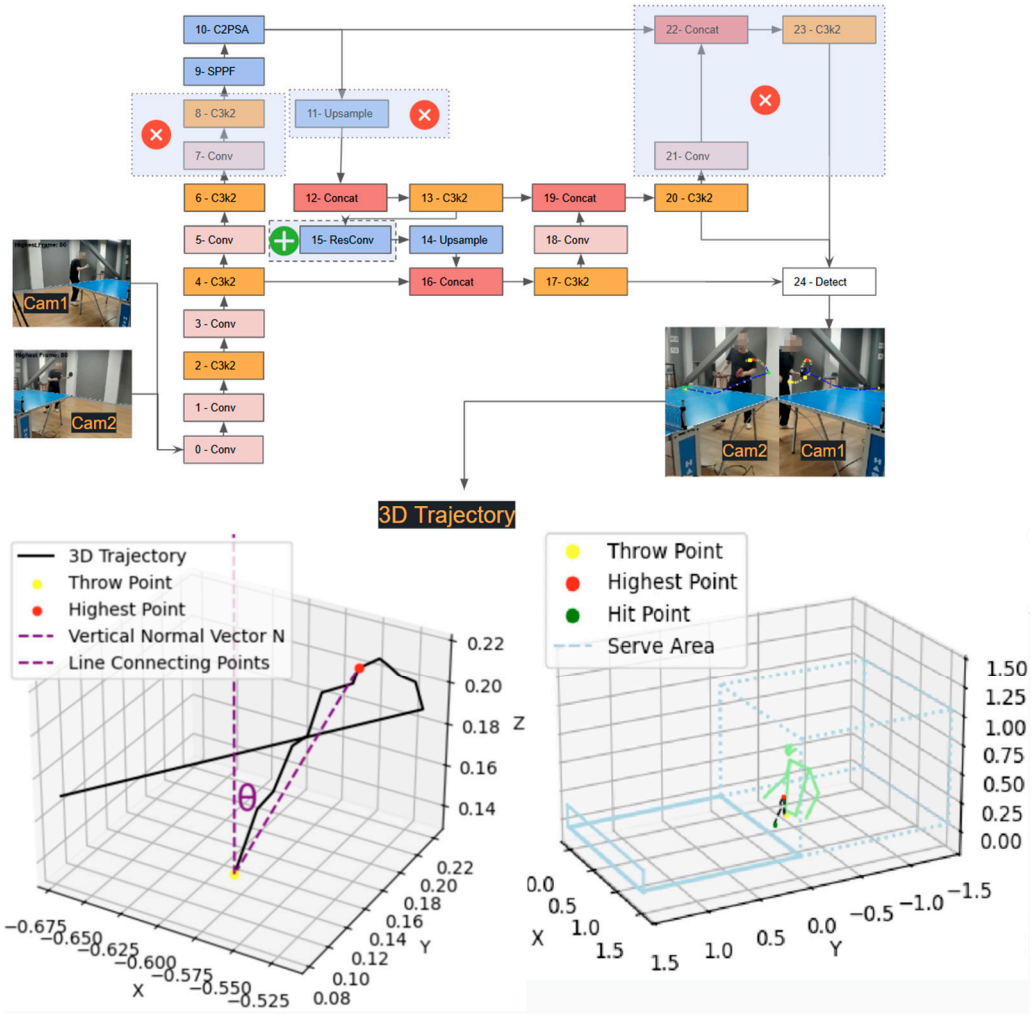


Figure 4. Modified YOLOv11 architecture for small object detection. The large object detection layer is removed, and a Custom *ResConv* layer is added to enhance detection performance for small, fast-moving objects like the table tennis ball.

The concatenated tensor x' 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 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 outcomes to reconstruct the ball 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 standard 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

To ensure high-quality video capture suitable for accurate ball tracking, the cameras were configured with specific settings to minimize motion blur. High shutter speeds of 1/500 s or faster were used during recording. These settings allowed us to capture sharp frames of the ball even during high-speed serves and smashes, where the ball can reach speeds of up to 20 m/s. This mitigation of motion blur ensured that the ball's position could be precisely detected in each frame, providing a reliable input for tracking algorithms.

ByteTrack was employed to maintain the ball position across consecutive frames [11]. It 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.

Figure 5 illustrates the basic workflow of ByteTrack in our system, where detection boxes are associated with existing tracks, and Kalman filtering helps maintain track consistency.

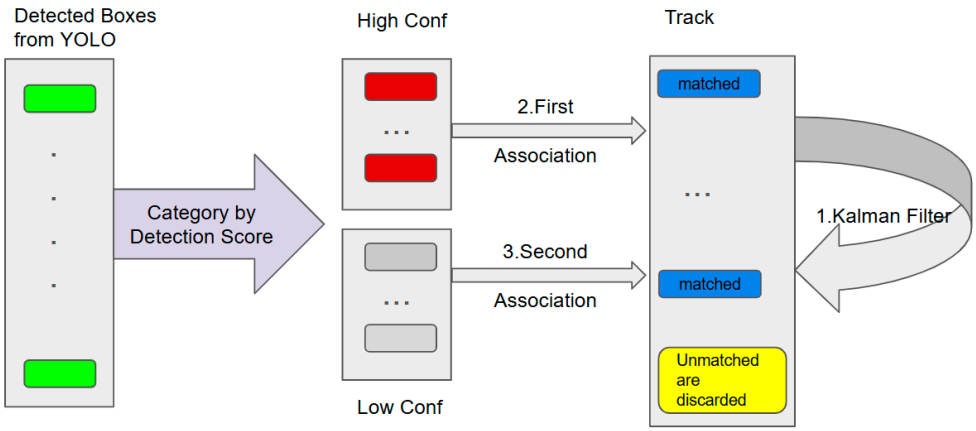


Figure 5. ByteTrack Framework for Table Tennis Ball Tracking starting from detection boxes output by YOLO. Detection boxes are categorized by confidence, with high-confidence boxes prioritized in the first association. Low-confidence detections are considered in a second association step, ensuring robust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

The background subtraction and optical flow methods [12] based on the OpenCV platform provide a simpler alternative for motion detection. However, they are susceptible to noise from lighting changes or background changes, which can cause track fragmentation. We will compare them with ByteTrack.

2.6. Video Segmentation

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

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) [13–16], the focus of this paper is exclusively on the ball 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 straightforward segmentation process relies on the spatial movement within the calibrated 3D space. A threshold $Y > 0.5$, representing a spatial boundary in the serve area, is utilized to identify the serve sequence. This criterion isolates the serve action by focusing on the ball trajectory during its active involvement in the serve, effectively capturing

the temporal and spatial characteristics of the sequence. Specifically, the Y-coordinate represents the ball's position in the forward–backward direction relative to the table, with the baseline of the player's side of the table set at Y = 0. The serve is considered complete when the ball crosses this threshold, marking the transition from the active serve motion to the post-serve phase.

2.7. Transformer Model for Key Point Detection

A Transformer model was applied to analyze 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 [17], which excels in capturing dependencies in sequential data, making it highly effective for recognizing patterns in the ball movement over time. By learning from labeled trajectories, the model can accurately predict critical points within the serve sequence, thereby enhancing the ability of this model to detect fouls based on serve dynamics.

The dataset used for training consists of labeled 3D trajectory data, specifically targeting key points in table tennis serves (throw point, highest point, and hit point). The data were collected from a series of table tennis serves, and they are structured in such a way that each trajectory consists of 3D coordinates (x, y, z) along with the corresponding frame indices. The dataset contains a total of 300 labeled serves, and it was split into 70% training, 15% validation, and 15% test sets. The training data were used to update the model's weights, the validation set was employed to tune hyperparameters, and the test data were used to evaluate the model's performance. The dataset includes serves from 3 different individuals with varied serve styles, ensuring that the model generalizes well across different scenarios.

Training was conducted with a batch size of 1, as each sample is a sequence of varying lengths depending on the serve. The model was trained for 100 epochs, utilizing CrossEntropyLoss for classification of the key points. Adam optimizer was employed with a learning rate of 0.001. These details provide a clearer understanding of the training setup and the dataset used.

The Transformer architecture has been widely used for modeling sequential data, particularly in natural language processing (NLP) tasks (Vaswani et al., 2017) [17]. For this study, we employed a modified version of the Transformer architecture, known as the TrajectoryTransformer, designed specifically for trajectory prediction. Unlike typical NLP tasks, where the input consists of word embeddings, our model processes sequential data consisting of 3D trajectory points (x, y, z) and the corresponding frame index. The input to the model was thus updated to 4 dimensions (3D coordinates + frame index).

Our approach is similar to those used in time-series prediction tasks, where Transformer models have been adapted to forecast sequential patterns in various domains, such as human motion prediction [16] and ball trajectory forecasting in sports analytics [3]. In particular, our model reduces the depth and complexity of the Transformer to improve efficiency for the real-time analysis of sports trajectories, using two encoder layers and four attention heads, which are optimal for our specific task.

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 precisely.

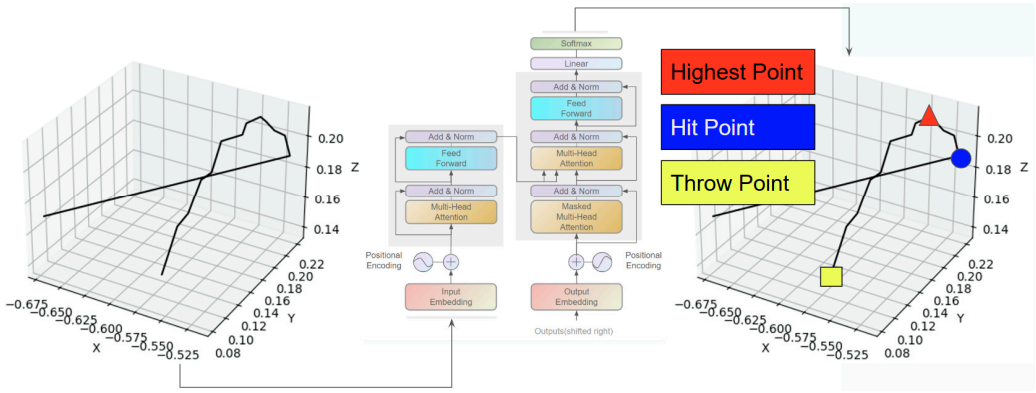


Figure 6. The left panel shows the trajectory path, while the right panel highlights the annotated turning points (throw point, highest point, and hit point). The Transformer model outlines its components, such as multi-head attention, which enables it to process the trajectory data and extract key turning points.

Like spatio-temporal Transformer networks (S2TNets) and the Trajectory Unified Transformer [18–20], Transformer-based models 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 is on the ball's motion alone, allowing the Transformer to concentrate on the trajectory patterns specific to the serve.

2.8. Rule-Based Foul Detection

A set of rule-based criteria is employed 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 [21].

Minimum Drop Height: To ensure that the serve meets the minimum height requirement, we monitor 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.
Vertical Angle at Throw: This assesses the initial toss direction. To ensure that the toss remains near the vertical axis, we calculate the angle between the vector connecting the throw point and the highest point of the ball trajectory and the vertical reference vector (0,0,1). This angle is computed by using Equation (2).

$$\theta = \arccos\left(\frac{A_x \cdot B_x + A_y \cdot B_y + A_z \cdot B_z}{\sqrt{A_x^2 + A_y^2 + A_z^2} \cdot \sqrt{B_x^2 + B_y^2 + B_z^2}}\right) \times \frac{180}{\pi} \quad (2)$$

where A_x, A_y, A_z are the components of the vector $\vec{A}$, which connects the throw point to the highest point, and B_x, B_y, B_z represent the components of the vertical reference vector $\vec{B} = (0, 0, 1)$. Equation (2) computes the angle in radians, which is then converted to degrees. If θ exceeds 30°, the toss is flagged as a potential foul for excessive backward tilt. As illustrated in Figure 7a, the system visualizes the ball's 3D trajectory, indicating key points such as the throw point (yellow) and the 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.
Service Area Positioning: The ball 3D coordinates are continuously tracked to ensure that the ball remains within the designated service area throughout the serve, as shown in Figure 7b. This spatial analysis prevents serves from originating outside the legal area, ensuring compliance with spatial regulations.

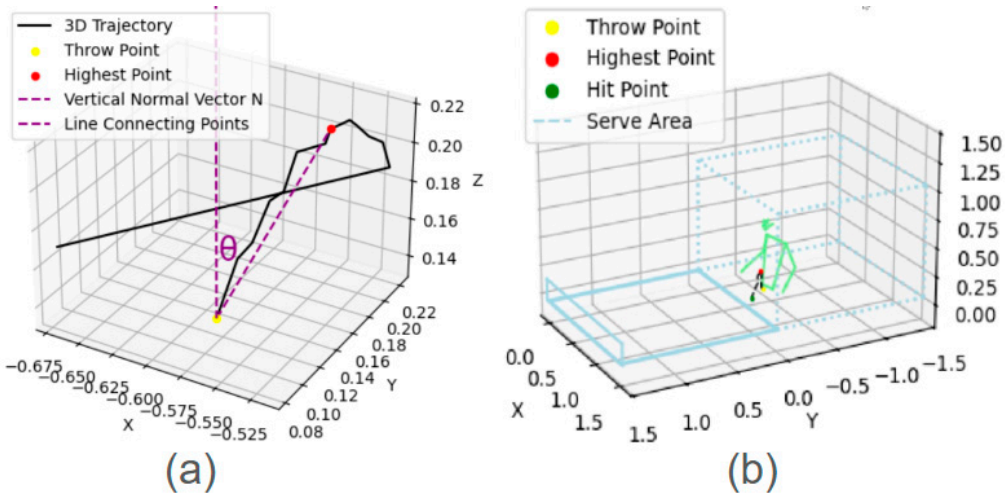


Figure 7. Rule-Based Foul Detection: (**a**) 3D trajectory visualization with the throw point (yellow), the highest point (red), and the calculated vertical angle. The vertical normal vector and the line connecting the key points are included for clear representation. (**b**) Spatial analysis of the serving area, showing the trajectory of the ball relative to the permitted boundaries and confirming compliance with service area rules.

3. Results

3.1. Implementation for Ball Detection

The performance was evaluated by comparing it with two benchmark models: YOLOv8 and YOLO11m. The evaluation focused on 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%, a mAP@50 of 86.87%, and a mAP@50:95 of 39.84%, outperforming both YOLOv8 and YOLO11m. The training time of our model was 4 h and 33 min, slightly longer than that of YOLOv8 but shorter than that of YOLO11m, reflecting a balance between accuracy and efficiency.

Table 1. Performance values of YOLOv8, YOLO11m, and this work.

	Precision	Recall	mAP@50	mAP@50:95	Training Time
YOLOv8	84.92%	75.40%	81.30%	36.30%	3 h 24 min
YOLO11m	81.25%	81.25%	84.12%	38.12%	4 h 59 min
This work	87.52%	83.37%	86.87%	39.84%	4 h 33 min

In the context of these metrics, mean average precision (*mAP*) serves as a critical evaluation indicator, measuring the average precision across different classes. mAP@50 denotes the *mAP* at a 50% threshold of intersection over union (IoU). Formally, the average precision (*AP*) for a specific class is computed as the area under the precision–recall curve:

$$AP = \int p(r)dr \tag{3}$$

The *mAP* is then derived as the mean of *AP* values across all classes:

$$mAP = \frac{1}{nc}\sum AP \tag{4}$$

nc is the total number of classes. Additionally, the stricter mAP@50:95 metric computes mAP values across IoU thresholds ranging from 50% to 95%, providing a more comprehensive evaluation of model performance.

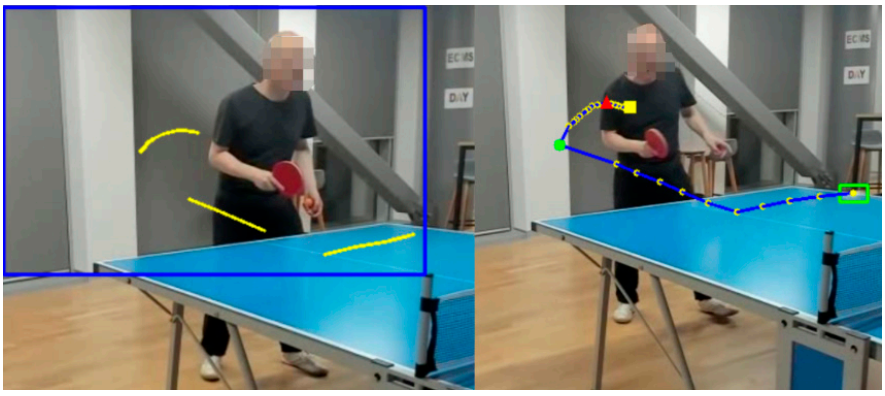
The progression of these metrics over 300 training epochs provides a detailed comparison of the models.

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 more significant 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. Our model shows steady improvement for the stricter mAP@50:95 metric, surpassing the performance of YOLOv8 and YOLO11m and confirming its reliability across a range of IoU thresholds.

The comparative analysis reveals that YOLOv8 benefits from a shorter training time but 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 achieves 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.

3.2. Implementation for Ball Tracking

The effectiveness of ball tracking methods in high-speed table tennis scenarios was evaluated by using two approaches: OpenCV background subtraction with optical flow and ByteTrack. Figure 8 illustrates each method's performance and 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 ByteTrack.



(**a**) (**b**)

Figure 8. Comparison of Ball Tracking Methods: (**a**) The OpenCV background subtraction and optical flow method result in fragmented and inconsistent ball trajectories due to background noise, rapid motion, and interference from moving players. (**b**) The ByteTrack method delivers smoother and continuous ball trajectories, robustly handles complex motion, and maintains tracking consistency.

Background subtraction and optical flow, implemented by using OpenCV, is relatively straightforward 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 paper, the ball's movement is effectively captured against a stable background, as shown in the left panel of Figure 8.

However, this approach exhibited a few limitations. It was susceptible to background noise, such as shadows and reflections, which often led to fragmented detection results.

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

3.3. Video Segmentation Results

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

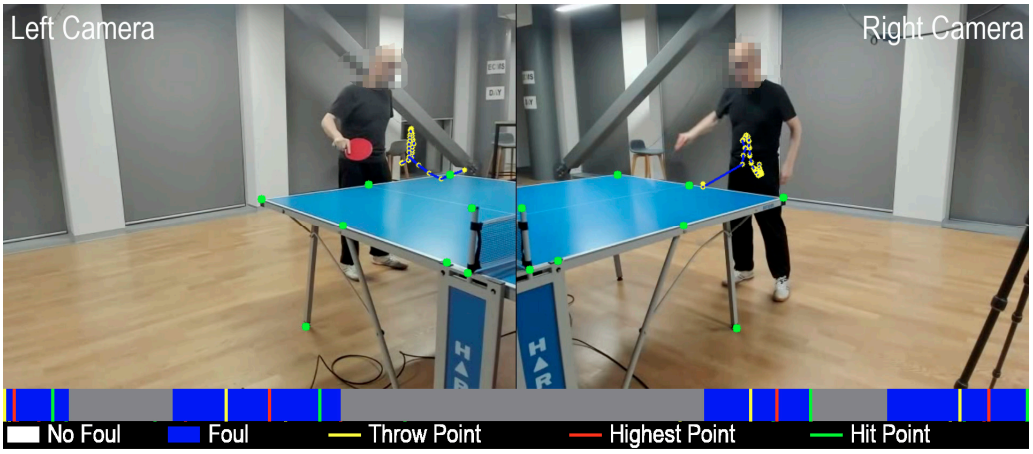


Figure 9. Video Segmentation Results: (**Top**) Synchronized left and right camera views showing key trajectory points during a table tennis serve. The throw point (yellow), highest point (red), and hit point (green) are marked to track the ball's trajectory. (**Bottom**) Segmented timeline showing frame classification as "No Foul" (gray) and "Foul" (blue), with transitions aligned to the serve events.

In Figure 9, the left and right camera views are synchronized to comprehensively analyze the serve. The 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 clearly visualizes the classification results. Blue segments indicate fouls, while gray segments represent no-foul frames. The system can seamlessly identify transitions between compliant and non-compliant actions.

The robust identification of critical trajectory points further validates the segmentation, ensuring that serve sequences are broken down into precise segments. By leveraging the ball trajectory and synchronized video data, the system achieves reliable segmentation 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.

3.4. Implementing Transformer Model for Key Point Detection

The results, as shown in Figures 10 and 11, 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.

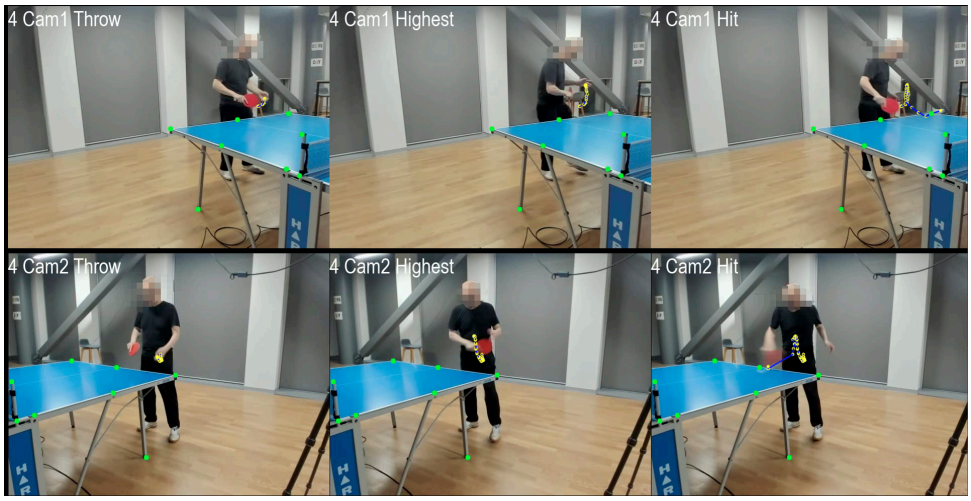


Figure 10. Multi-Camera Frames of Turning Points: Frames from Cam1 and Cam2 correspond to three key moments: the throw point, the highest point, and the hit point, as identified by the Transformer model. The high F1 score (0.93) demonstrates the accuracy of detecting these critical moments.

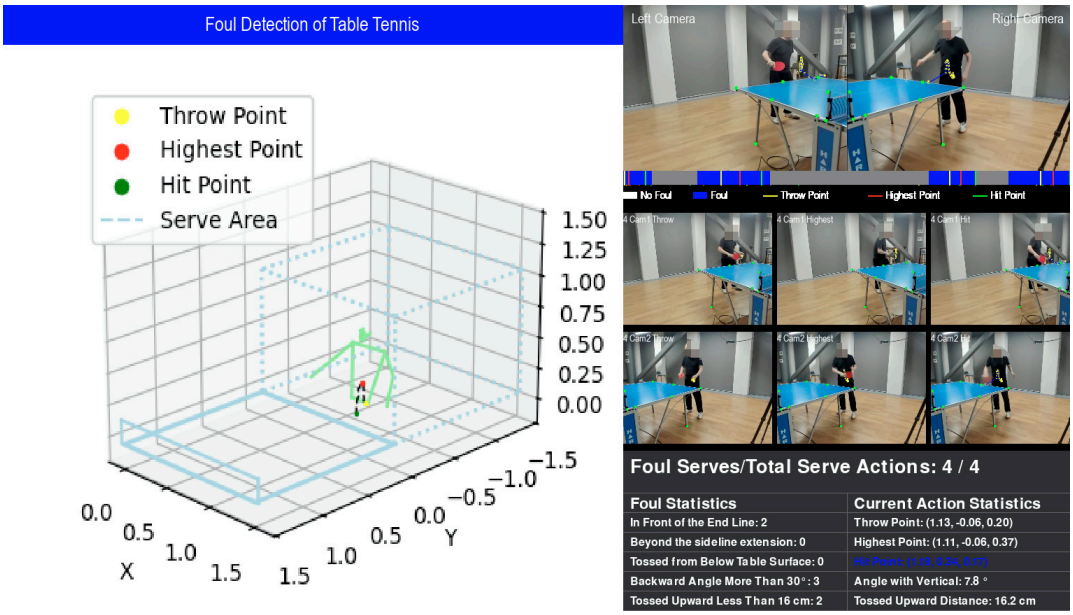


Figure 11. The 3D Trajectory and Serve Statistics: The central 3D trajectory plot highlights the ball motion and key turning points aligned with the spatial limits of the serving area (dotted boundary box). The right side provides serve statistics, showing the timeline of compliant and non-compliant frames, foul counts, and detailed metrics for the current serve, including tossed upward distance and the angle with the vertical axis.

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 10, showing synchronized views for the detected turning points. Our verification of these turning points yielded an F1 score of 0.93, confirming the accuracy of the prediction and its reliability in detecting key moments during the serve.

The F1 score is a widely used evaluation metric that balances precision and recall, making it particularly useful for classification tasks like key point detection. It is defined as:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

where precision is the proportion of true positive detections among all detected points (i.e., how many of the predicted key points were correct). Recall is the proportion of true positive detections among all true key points (i.e., how many of the actual key points were detected).

In our case, the high F1 score of 0.93 indicates that the model's key point predictions were both precise and comprehensive, making it a reliable tool for identifying critical moments in the serve action.

Figure 11 provides a comprehensive analysis of turning points within the 3D trajectory and foul detection context. The central 3D trajectory plot visually illustrates the ball motion, with key turning points (throw, highest, and hit points) highlighted. The plot also shows the 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 and hit points fall within the boundary, ensuring the serve adheres to spatial constraints.

The results of the experiment, as shown in Figure 11, 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), visually representing 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.

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 using a multi-camera setup specifically designed for high-speed motion tracking. The dataset was employed 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.

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 h and 33 min. 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 is further tailored for different usage scenarios.

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, these 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 maintains high segmentation accuracy and reduces computational demands, making it well-suited for real-time applications in table tennis.

The Transformer model performed exceptionally well, achieving an F1 score of 0.93 in manually validating 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 are pivotal in determining compliance with game regulations. However, while the results are promising, additional experiments with more extensive and diverse datasets are necessary to validate the model's generalizability across different playing conditions and ball trajectories.

The trajectories shown in Figures 6 and 7 exhibit some non-smooth behavior, which is primarily due to two factors: detection inaccuracies and tracking inconsistencies. While the YOLO model generally performs well, certain scenarios, such as high-speed movements, occlusions, or background clutter, can cause missed detections or false positives. These errors lead to small fluctuations in the trajectory, causing abrupt changes in direction or velocity. Additionally, the ByteTrack algorithm, which tracks the ball across frames, occasionally experiences temporary loss of track or incorrect associations during fast movements or occlusions, further contributing to the non-smooth trajectories.

In Figure 8, although the trajectories appear smoother, minor inconsistencies still occur due to occasional occlusions or changes in lighting conditions that affect detection accuracy. These factors result in slight disruptions to the continuity of the ball's movement.

Despite the promising results, the project has a few 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.

Our 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. Conclusions

In this paper, we developed a serve foul detection system for table tennis games 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 achieved an F1 score of 93% when detecting turning points, such as the throw, highest, and hit points. We effectively segment the serve sequence by using only the ball 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. and W.Q.Y.; data collection, G.L.Y. and W.Q.Y.; writing—original draft preparation, G.L.Y.; writing—review and editing, G.L.Y., W.Q.Y., M.N. and X.J.L.; visualization, G.L.Y. and W.Q.Y.; supervision, W.Q.Y. and M.N.; project administration, W.Q.Y. and M.N.; funding, W.Q.Y., M.N. and X.J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research has no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts 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. [CrossRef]
2. Kulkarni, K.M.; Jamadagni, R.S.; Paul, J.A.; Shenoy, S. Table Tennis Stroke Detection and Recognition Using Ball Trajectory Data. *arXiv* **2023**, arXiv:2302.09657. [CrossRef]
3. Voeikov, R.; Falaleev, N.; Baikulov, R. TTNet: Real-time temporal and spatial video analysis of table tennis. *arXiv* **2020**, arXiv:2004.09927. [CrossRef]
4. Huang, Y.-C.; Liao, I.-N.; Chen, C.-H.; İk, T.-U.; Peng, W.-C. TrackNet: A Deep Learning Network for Tracking High-speed and Tiny Objects in Sports Applications. *arXiv* **2019**, arXiv:1907.03698. [CrossRef]
5. Caio, M.D.; Van Zandycke, G.; De Vleeschouwer, C. Context-Aware 3D Object Localization from Single Calibrated Images: A Study of Basketballs. In Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports, Ottawa, ON, Canada, 29 October 2023; pp. 49–54. [CrossRef]
6. Hung, C.-H. A Study of Automatic and Real-Time Table Tennis Fault Serve Detection System. *Sports* **2018**, *6*, 158. [CrossRef]
7. Yan, W.Q. *Computational Methods for Deep Learning: Theory, Algorithms, and Implementations*; Springer Nature: Berlin/Heidelberg, Germany, 2023; pp. 154–196.
8. Poliakov, A.; Marraud, D.; Reithler, L.; Chatain, C. Physics Based 3D Ball Tracking for Tennis Videos. In Proceedings of the International Workshop on Content Based Multimedia Indexing (CBMI), Grenoble, France, 23–25 June 2010; pp. 1–6. [CrossRef]
9. 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. [CrossRef]
10. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of YOLO Algorithm Developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
11. 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, Switzerland, 2022; pp. 1–21. [CrossRef]
12. Barron, J.L.; Fleet, D.J.; Beauchemin, S.S. Performance of Optical Flow Techniques. *Int J Comput Vis.* **1994**, *12*, 43–77. [CrossRef]
13. Ding, G.; Sener, F.; Yao, A. Temporal Action Segmentation: An Analysis of Modern Techniques. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 1011–1030. [CrossRef] [PubMed]
14. 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. Multimed. Comput. Commun. Appl.* **2024**, *20*, 1–23. [CrossRef]
15. Tran, T.-D. TTNet: A Novel Machine Learning Model for Facial Emotion Detection in Online Learning Systems. *SoftwareX* **2024**, *27*, 101787. [CrossRef]
16. 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, Virtual, 20 October 2021; MMSports'21. Association for Computing Machinery: New York, NY, USA, 2021; pp. 35–41. [CrossRef]
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; ukasz Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Nice, France, 2017; Volume 30.
18. Hu, S.; Shen, L.; Zhang, Y.; Chen, Y.; Tao, D. On Transforming Reinforcement Learning With Transformers: The Development Trajectory. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 8580–8599. [CrossRef] [PubMed]
19. 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, Virtually, 17–19 November 2021; pp. 454–469.

20. 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), Paris, France, 2–3 October 2023; pp. 9675–9684.
21. 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. *J. Coach. Educ. Sports* **2024**, *5*, 25–32. [CrossRef]