TrackNetV3: Enhancing ShuttleCock Tracking with Augmentations and Trajectory Rectification

Yu-Jou Chen
National Yang Ming Chiao Tung University
qaz812345@gmail.com

Yu-Shuen Wang
National Yang Ming Chiao Tung University
yushuen@cs.nycu.edu.tw

ABSTRACT

We present TrackNetV3, a sophisticated model designed to enhance the precision of shuttlecock localization in broadcast badminton videos. TrackNetV3 is composed of two core modules: trajectory prediction and rectification. The trajectory prediction module leverages an estimated background as auxiliary data to locate the shuttlecock in spite of the fluctuating visual interferences. This module also incorporates mixup data augmentation to formulate complex scenarios to strengthen the network's robustness. Given that a shuttlecock can occasionally be obstructed, we create repair masks by analyzing the predicted trajectory, subsequently rectifying the path via inpainting. This process significantly enhances the accuracy of tracking and the completeness of the trajectory. Our experimental results illustrate a substantial enhancement over previous standard methods, increasing the accuracy from 87.72% to 97.51%. These results validate the effectiveness of TrackNetV3 in progressing shuttlecock tracking within the context of badminton matches. We release the source code at https://github.com/qaz812345/TrackNetV3.

CCS CONCEPTS

• **Computing methodologies** → **Tracking**; **Object detection**.

KEYWORDS

Badminton, Shuttlecock tracking, trajectory rectification

ACM Reference Format:
Yu-Jou Chen and Yu-Shuen Wang. 2023. TrackNetV3: Enhancing ShuttleCock Tracking with Augmentations and Trajectory Rectification. In *ACM Multimedia Asia 2023 (MMAsia '23), December 6–8, 2023, Tainan, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3595916.3626370

Figure 1: Given the small size of shuttlecocks and their tendency to blend into backgrounds, tracking them can be challenging. For clarity, we have marked the shuttlecock positions using red circles. Note that our TrackNetV3 can successfully detect the shuttlecocks in these two examples.

1 INTRODUCTION

The exponential growth of deep learning research in recent years has led to its widespread application in numerous domains, one of which is sports analytics. With the advent of the internet, abundant game footage is now available for in-depth analysis. Machine learning has become a key player in this field, paving the way for automated systems that enable efficient game analysis [1, 4, 7, 24, 28].

In this context, our work is dedicated to the tracking of shuttlecocks in badminton, a sport with a global footprint that requires its participants to engage in strategic planning.

The unique attributes of a shuttlecock, including its lightweight nature and rapid motion, present particular challenges for tracking. Firstly, during a smash, a shuttlecock can achieve instantaneous peak velocities reaching 426 km/h, making it the fastest moving object in all racquet sports. This rapid movement often leads to motion blur when captured by standard cameras, causing a ghosting effect of the shuttlecock. Secondly, the wide-angle perspective of court view shots usually reduces the shuttlecock to a minuscule portion of the frame, further complicating the tracking task. Thirdly, a shuttlecock can blend seamlessly into the background components, such as the net and court lines, rendering it nearly invisible if its dynamics are not analyzed through consecutive video frames.

MMAsia '23, December 6–8, 2023, Tainan, Taiwan

ACM ISBN 979-8-4007-0205-1/23/12…$15.00
https://doi.org/10.1145/3595916.3626370

Given these challenges, our goal is to develop a more precise shuttlecock tracking system. The suggested TrackNetV3 consists of two fundamental modules: trajectory prediction and rectification. Building on the foundation of TrackNetV2, the trajectory prediction module considers a series of video frames to generate corresponding heat maps that indicate the positions of the shuttlecock. We further enrich this model by feeding it an estimated background, which acts as supplemental information to better differentiate the shuttlecock from various visual distractions. The trajectory prediction module also incorporates mixup data augmentation, creating complex scenarios that enhance the network's robustness. Recognizing that a shuttlecock may occasionally be concealed, making its true position undeterminable through visual cues, we rectify the trajectory based on the shuttlecock's motion dynamics. Specifically, we generate repair masks by assessing the predicted trajectory and train the rectification module that considers the predicted trajectory and the mask for rectification. This process considerably enhances the accuracy of tracking and the completeness of the trajectory.

Our system was assessed against advanced techniques like YOLOv7 and TrackNetV2. The results of these comparative experiments demonstrate that TrackNetV3 significantly outperforms these baseline methods, yielding accuracies of 97.51%, contrasted with 94.98% for TrackNetV2 and 53.47% for YOLOv7. Furthermore, we conducted an ablation study to evaluate the presented tracking strategies. This study corroborates the efficacy of the proposed strategies in tracking swift and minuscule shuttlecock trajectories in broadcast badminton videos.

2 RELATED WORKS

Deep learning models have been widely used in tracking balls across various sports, encompassing tennis, golf [23, 29], and badminton [12, 25]. This task shares considerable overlap with object detection applications and can be executed using conventional methods like R-CNN (regions with convolutional neural network features) [8, 9, 21] and YOLO (you only look once) [18]. Specifically, R-CNN employs a region proposal network (RPN) to recognize objects and suggest numerous regions of interest (ROIs) as candidates. These ROIs undergo categorization and localization, producing the final category and bounding box for all objects. YOLO approaches detection as a regression problem. It divides the image into an $N \times N$ grid, predicting multiple results centered on each cell. Overlapping results are discarded using confidence scores and Non-Maximum Suppression. YOLO has experienced several iterations [3, 18–20, 26], with the latest being YOLOv7 [26], which employs several methods relating to model re-parameterization, dynamic label assignment strategy, and model scaling. These enhancements have minimized the model's necessary parameters and computational demand, thereby improving speed and accuracy.

In terms of object tracking [2, 5, 11, 14, 15, 27, 30], the standard procedure commences by defining the target object's position in the initial frame. A search radius is established around the target, and the model matches features within this radius to ascertain the object's new position at the next temporal point. This updating of the search radius and object location persists until the end of a video. The most common approach involves the use of Siamese networks [6], typically comprising two feature extractors that share weights.

Figure 2: The left image represents the median frame derived from a rally. On the right, we showcase a cleaner background, which we ascertain by calculating the median of such median frames from an entire match.

They separately extract the features of the target and the search area and compare them to pinpoint the target's location. SiamFC [2], the pioneer in using Siamese networks for object tracking, utilized a fully convolutional network [17] structure to extract features and compute cross-correlation to gauge the similarity between target and background features. SiamBAN [5] proposed a box adaptive head capable of predicting bounding boxes directly, eliminating the need for additional anchor boxes.

Despite the proven effectiveness of the YOLO series in object detection and tracking, challenges arise when dealing with balls, as they are small objects that can blur and produce ghost images when moving at high speeds. This makes it challenging for standard detection models to identify their location based on static images alone accurately. For tracking models, the ball's velocity might surpass the estimated search radius, causing the model to lose sight of its target. Furthermore, significant distortions caused by motion blur could interfere with feature similarity matching, resulting in an inability to specify targets. In response to these issues, TrackNet [12] focuses on tracking badminton shuttlecocks and generates a corresponding heatmap for each video frame, indicating the shuttlecock's position. By taking multiple frames into account simultaneously, this approach enables the model to learn the features of the shuttlecock and its trajectory, counteracting difficulties in detection and tracking due to the shuttlecock's diminutive size. TrackNetV2 [25] employed the U-Net architecture [22]. The skip connections fused low-level and high-level features, improving tracking precision. They also introduced weighted binary cross entropy to handle class imbalances in the heat maps.

Building on TrackNetV2 [25], the introduced TrackNetV3 utilizes an estimated background as supplemental data to pinpoint the shuttlecock's location. It also incorporates mixup data augmentation to establish intricate scenarios, thereby bolstering the network's robustness. Furthermore, since a shuttlecock can occasionally be concealed, we generate repair masks by assessing the forecasted trajectory. We then rectify the path through inpainting. The integrated strategies considerably improve the precision of tracking.

3 METHOD

The presented TrackNetV3 comprises a trajectory prediction module and a rectification module. The architecture of our tracking module (see Figure 3 (a)) is designed with a U-Net structure, incorporating convolution layers and skip connections. The inputs to this network are concatenated video frames and an estimated

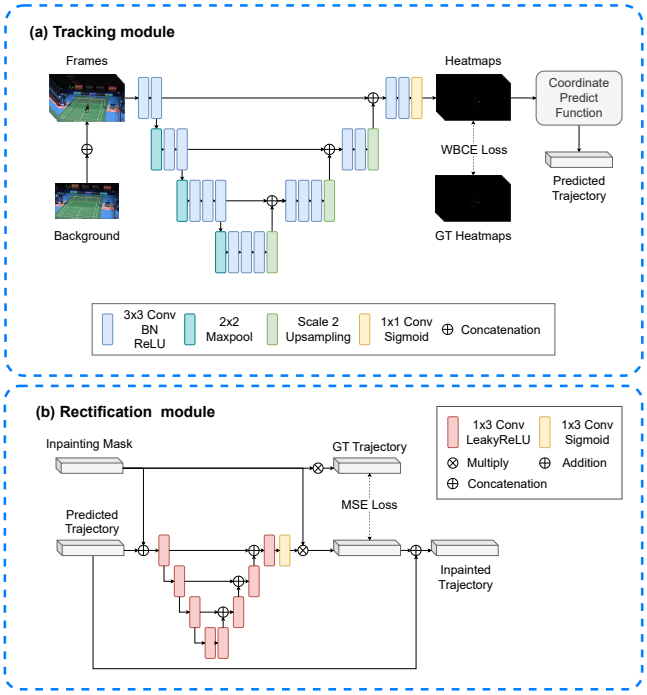

Figure 3: The network architecture of TrackNetV3. (a) The tracking module processes a sequence of video frames to produce respective heat maps, identifying the shuttlecock locations. (b) The rectification module refines shuttlecock trajectories by filling in erroneously detected shuttlecock positions.

background image. The outputs are a series of heat maps, each corresponding to a frame and indicating the probable location of the shuttlecock. A higher heat map value indicates a higher likelihood of the shuttlecock's presence. Since most heat map areas are nearly zero, the trained networks are prone to degenerate and produce heat maps with universally zero values. To mitigate this imbalance problem, we adopt the strategy from TrackNetV2 and train the network by minimizing the weighted binary cross-entropy loss.

$$L_{wbce} = -\sum_{i=1}^{n}(1-w)^2\hat{y}_i \log y_i + w^2(1-\hat{y}_i)\log(1-y_i), \quad (1)$$

where y_i and $\hat{y}_i$ denote the predicted and ground truth heat map values, respectively. We set $w = y_i$ to amplify the contribution of incorrectly predicted pixels, akin to the focal loss strategy [16]. After training, the shuttlecock's position was estimated by binarizing the heat map using a threshold of 0.5 and calculating the centroid of the largest separate blob.

Our tracking module's training process utilizes the heat map generation method proposed in TrackNet. Specifically, we determine the shuttlecock's average size in match videos and dynamically generate heat maps based on the labeled shuttlecock positions. Pixel values within the radius of the shuttlecock coordinates are set to 1 as our training targets. The values are set to 0 otherwise.

3.1 Background Estimation

Taking advantage of the fixed viewpoint from broadcast cameras during matches, we estimate the background by computing the median frame from a court view shot video. That is, each pixel value is determined by the median of the values along the temporal axis. In our dataset, a match is divided into several scoring segments or "rallies". Intuitively, estimating the background by computing the median frame within each rally seems accurate. However, if a player stays still for an extended period during serve preparation within a rally, the estimated background might include a residual image of the player, creating a "ghosting" effect. To address this, we use a "match median frame" strategy. We calculate the median frame from each rally within the same match, and then compute the median of these results, providing a better background estimate.

It is worth noting that there are several strategies for utilizing the estimated background information. The simplest—and experimentally the most effective—strategy is to concatenate the background image when feeding video frames into the tracking module. We also experimented with other strategies, such as replacing the video frames with the magnitudes of background subtraction frames, and concatenating these magnitudes. We discuss the experimental results of these strategies in Section 4.5.

3.2 Mixup Augmentation

Mixup augmentation is a technique that encourages model robustness by creating synthetic training examples. It generates new samples by taking a convex combination of pairs of inputs and their corresponding labels. In mathematical terms, for inputs x_1, x_2 and their labels y_1, y_2, mixup creates a new input x' and its label y' as follows: $x' = \lambda x_1 + (1-\lambda)x_2$ and $y' = \lambda y_1 + (1-\lambda)y_2$. Here, λ is a random number between 0 and 1 drawn from a beta distribution.

In the process of shuttlecock tracking, we randomly trim two video clips of identical length from court view shot videos. With mixup augmentation, we generate a new video clip and its corresponding heat maps. While this approach may appear counterintuitive, we discovered through experimentation that it's more effective than creating synthetic frames in a video clip, where the shuttlecock occupies a "blend" of positions. In essence, the mixing of video frames fails to enhance the model's resilience to the blurriness of the shuttlecock position. We will elaborate on the experiment results in Section 4.5.

3.3 Shuttlecock Trajectory Rectification

In badminton games, there are instances where the shuttlecock may overlap with sponsor advertisements or be obscured by players or the net, making it difficult to identify. In such cases, all values on heat maps would be near zero. Our solution is to interpolate the missing shuttlecock coordinates based on the currently predicted trajectory. Borrowing from the concept of image inpainting, we define an "inpainting mask" to identify potential frames that require correction. We then train a trajectory rectification module that takes this mask and the predicted trajectory from the tracking module as inputs, generating a corrected trajectory as output.

The model's failure to predict the shuttlecock's presence can be ascribed to three scenarios: occlusion by an object, extreme difficulty to discern visually, or the shuttlecock simply flying out of the

camera's field of view. As the shuttlecock is only considered existent when within the field of view, we establish a height threshold as a determinant to decide whether unseen sequences necessitate inpainting. Specifically, for a video frame i where the shuttlecock is undetected, we track frame i forward and backward until we reach the frames where the shuttlecocks are detected. Let $[f+1, b-1]$ be the interval of video frames where the shuttlecock is missing, where $f < i < b$. We generate the inpainting masks according to the formula provided below:

$$M_i = \begin{cases} 1 & \text{if } p_y^f < \delta \text{ and } p_y^b < \delta \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where p_y^f and p_y^b represent the positions of the shuttlecock at frames f and b, respectively. In our model, we set δ to 30 pixels. Note that $M_i = 0$ if the shuttlecock at frame i is successfully detected.

The rectification module adopts a U-Net architecture with the inputs and outputs being the predicted and repaired 2D shuttlecock trajectories, respectively. These predicted trajectories are supplemented with a mask that highlights the frames that the rectification module is tasked with repairing. The architecture is outlined in Figure 3 (b). To process trajectory data, 1D convolution layers are employed, with kernels spanning in the temporal coordinate. This network is trained by minimizing the mean square error. During training, we generate a random mask where approximately 30% of the frames (i.e., M_i) are set to 1. In the inference phase, the mask M_i is determined using the formula referenced in Equation 2.

3.4 Implementation Details

Our tracking module considers multiple video frames simultaneously (specifically, 8 frames in our implementation, which demonstrated superior performance in our experiments) to identify the shuttlecock's location. By adopting overlapping sampling across the full sequence of frames, each frame can yield a number of heat map predictions equivalent to the length of the input. We assemble these predicted outcomes through a weighted averaging technique. Given that frames in the middle of the sequence possess a more comprehensive context, we surmise that the central predictions are more dependable. Consequently, we attribute weights to these predictions based on a Gaussian kernel, giving higher weights to central predictions and lower weights to those on the peripheries of the sequence. The final prediction is then acquired by calculating the weighted sum of all heat maps.

We used the Adam optimizer [13] with a learning rate set at 10^{-3} to train the tracking and rectification networks. The network's hyperparameters were initialized using the Xavier method [10]. The batch sizes for the tracking and rectification modules were set at 10 and 32, respectively. The training process lasted for 30 epochs, and we chose the model instances that yielded the highest validation accuracies.

4 RESULTS AND EVALUATIONS

The performance of TrackNetV3 was evaluated through both a comparative and an ablation study. The details are as follows.

Method	Accuracy	Precision	Recall	F1	FPS
YOLOv7	57.82%	78.53%	59.96%	68%	34.77
TrackNetV2	94.98%	**99.64%**	94.56%	97.03%	27.70
TrackNetV3	**97.51%**	97.79%	**99.33%**	**98.56%**	25.11

Table 1: Quantitative evaluation and timing statistics for baseline methods and our proposed TrackNetV3. The highest-performing results are denoted in bold fonts.

4.1 Dataset

We evaluated our model on the dataset released by TrackNetV2 [25], comprising court view shot videos of badminton games with a 720×1280 resolution. In this dataset, each match is divided into multiple rallies, with each frame having manually labeled shuttlecock positions. The training set includes 23 professional matches, totaling 172 rallies, and the test set comprises three professional matches, totaling 29 rallies. For experimental purposes, we carved out one rally from each match in the training set to formulate a validation set for model optimization. We also down-sampled the videos to a resolution of 288×512 during evaluation [25].

4.2 Evluation Metrics

We evaluated the performance of our network by calculating the distance between the predicted and the actual positions of the shuttlecock. Following the work of TrackNetV2 [25], if the distance is within 4 pixels, the detection is deemed accurate. Conversely, a distance greater than 4 pixels or inconsistency in the detection of the shuttlecock's presence or absence is deemed inaccurate. In addition to accuracy, we calculated the precision, recall, and F1 score for further evaluation. We also recorded the frames per second (FPS) to assess the efficiency of our system. We opted not to use the widely adopted mean average precision (mAP) metric due to the small size of the shuttlecocks. The area intersection over Union (IoU) employed in this metric tends to be highly unstable. As noted in the work of [29], a 3×3 pixel bounding box would experience a 50% decrease in IoU with just a single pixel offset.

4.3 Comparison to Baseline Methods

We assessed TrackNetV3's performance by comparing it with two benchmark methods. Firstly, we chose YOLOv7 (specifically the YOLOv7-X version[1]), a representative of single-image detection models, which was pretrained on the MS COCO Dataset and fine-tuned on the shuttlecock trajectory dataset. Secondly, we compared it with our implemented TrackNetV2. Although an official implementation of TrackNetV2 is available[2], we chose our implementation as a benchmark for comparison due to its enhanced performance (the original study reported an accuracy of 87.72%, whereas our implementation achieved 94.98%).

Figure 4 and Table 1 show the comparative results. As indicated, YOLOv7 generally performs less favorably than the TrackNet series, given the highest miss rate (i.e., 1-recall), whereas TrackNetV3, with its trajectory repair function, yields the lowest. Both YOLOv7 and TrackNetV2 struggle with detecting specific challenging samples,

[1] https://github.com/WongKinYiu/yolov7
[2] https://nol.cs.nctu.edu.tw:234/open-source/TrackNetv2

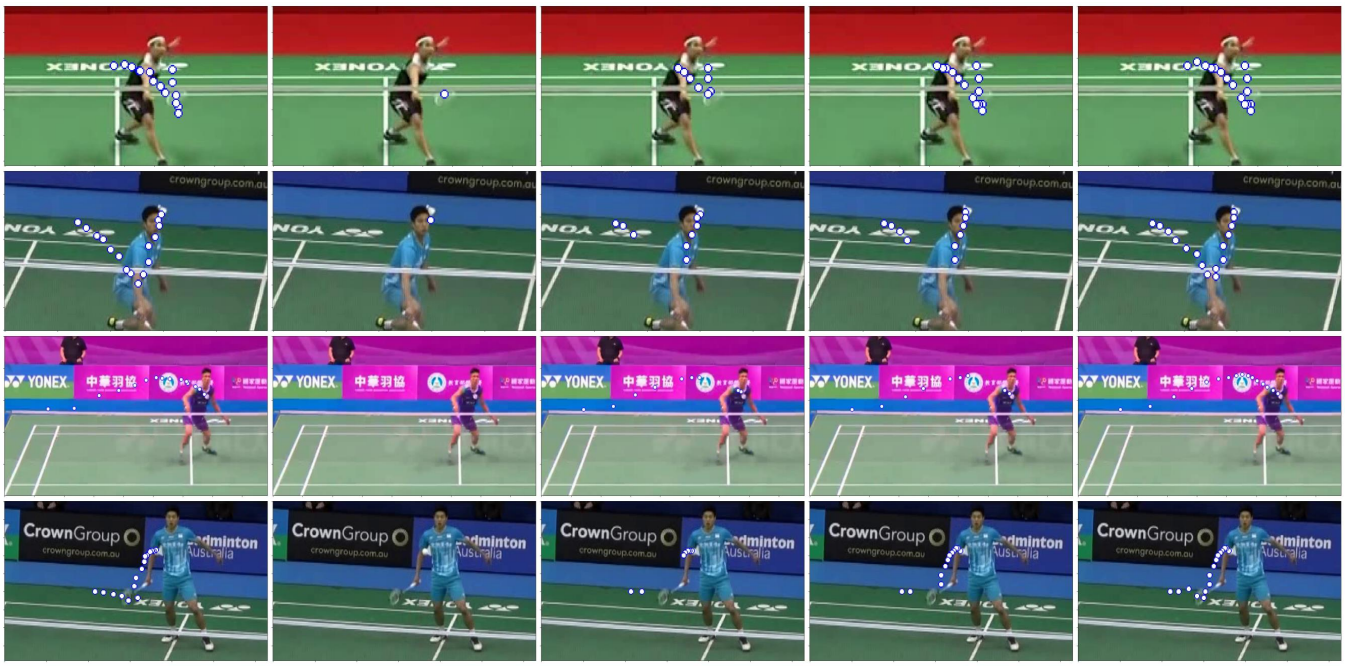

Figure 4: We conducted a comparative analysis of tracking results achieved using YOLOv7, TrackNetV2, and TrackNetV3, both with and without rectification modules. Each example portrays a video frame with previously tracked shuttlecock positions, which are represented by white dots surrounded by blue boundaries. As observed, YOLOv7 overlooked numerous shuttlecocks. Although TrackNetV2 demonstrated superior detection efficacy, it underperformed TrackNetV3.

especially when shuttlecocks overlay players or backgrounds, as shown in Figure 4. TrackNetV2 achieved the highest precision, suggesting that despite TrackNetV3's ability to repair the trajectory based on the shuttlecock's motion, the repaired positions may not always align closely with the shuttlecock's actual positions. This trajectory repair process improves recall but slightly compromises precision. We discuss this aspect in more detail in Section 4.4. Lastly, YOLOv7 has the highest processing speed, but the TrackNet series also operate close to real-time, even when they ensemble eight tracking results in the experiment. Notably, without the ensembling strategy, they could process over 75 frames per second, with a marginal decrease in accuracy of approximately 0.2%.

4.4 Ablation Study

We have enhanced TrackNetV2 by utilizing an estimated background image, implementing mixup data augmentation, and correcting trajectories. We undertook an ablation study to gauge the individual impact of these strategies. The outcomes of this study are detailed in Table 2. In this analysis, we report the values of true positives (TP), true negatives (TN), two types of false positives (FP1 and FP2), and false negatives (FN). FP1 represents instances where the detected shuttlecock isn't sufficiently close to the ground truth position, while FP2 indicates instances where a detected shuttlecock is falsely identified, i.e., no actual shuttlecock is present. The results show a clear, incremental improvement in tracking accuracy with each strategy incorporated. However, including the inpainting strategy led to a noticeable increase in FP1 and FP2. As the rectification module is designed to predict the positions of nearly unseen shuttlecocks, the predicted positions may deviate from the shuttlecock's actual positions (leading to FP1). Moreover, it sometimes predicts the shuttlecock positions even when the shuttlecocks are occluded, which are unlabeled in the dataset (contributing to FP2).

4.5 Discussions

Background Utilization. We used the estimated background image to assist the tracking module in distinguishing foreground objects. Although this background image remains the same throughout the match, it has proven effective in enhancing tracking performance (Table 2). However, we questioned whether there might be more effective ways to utilize this background information. Specifically, we experimented with two additional strategies. First, we subtracted the background image from each video frame, resulting in a map of pixel magnitudes. Second, we concatenated the magnitude maps with the original video frames to form the input. As shown in Table 3, when not using mixup augmentation, the second strategy (frame $\oplus$ Sub) yielded the best performance. However, its effectiveness significantly decreased when mixup augmentation was applied. This suggests that different strategies for incorporating background information may have varying interactions with other methods used in the tracking module, such as data augmentation.

Method	Accuracy	Precision	Recall	F1	TP	TN	FP1	FP2	FN
TrackNetV2	94.98%	99.64%	94.56%	97.03%	82.07%	**12.91%**	0.26%	**0.04%**	4.72%
+ Background	95.64%	99.37%	95.58%	97.44%	82.83%	12.81%	0.39%	0.14%	3.83%
+ Mixup	96.12%	**99.62%**	95.90%	97.73%	83.25%	12.87%	**0.24%**	0.07%	3.56%
+ Rectification	**97.51%**	97.79%	**99.33%**	**98.56%**	**84.99%**	12.51%	1.49%	0.43%	**0.57%**

Table 2: We assess the effectiveness of strategies used, including background concatenation, mixup data augmentation, and trajectory rectification. Note that the final row is representative of TrackNetV3, and FN is equivalent to the miss rate.

Background	Mixup	Accuracy	Precision	Recall	F1
Sub	No	91.89%	99.18%	91.39%	95.12%
frame ⊕ BG		95.64%	99.37%	95.58%	97.44%
frame ⊕ Sub		96.04%	99.50%	**95.91%**	97.68%
frame ⊕ BG	Sample	**96.12%**	**99.62%**	95.90%	**97.73%**
frame ⊕ Sub		95.94%	99.58%	95.72%	97.61%
frame ⊕ BG	Frame	95.00%	98.97%	95.21%	97.06%
frame ⊕ Sub		94.24%	98.39%	94.91%	96.62%

Table 3: We analyze the efficacy of background image utilization and mixup data augmentation. In this analysis, the rectification module was not utilized. 'Frame' and 'BG' denote the original video frame and the estimated background image, respectively. The symbol ⊕ represents the concatenation operator, and 'Sub' signifies the magnitude of subtraction between 'Frame' and 'BG'. The results reveal that the combination of video frames with the estimated background image, along with the employment of sample mixup augmentation, yields the highest tracking performance.

Mask Ratio	Accuracy	Precision	Recall	F1
0.2	97.23%	97.47%	**99.33%**	98.39%
0.3	**97.51%**	**97.79%**	**99.33%**	**98.56%**
0.4	97.48%	97.76%	**99.33%**	98.54%

Trajectory length	Accuracy	Precision	Recall	F1
8	97.33%	97.59%	**99.33%**	98.45%
16	**97.51%**	**97.79%**	**99.33%**	**98.56%**
32	96.41%	96.53%	99.32%	97.91%

Table 4: The rectification module's performance is robust to changes in mask ratios and trajectory lengths during training. We selected a ratio of 0.3 and a length of 16 due to the optimal performance in shuttlecock tracking.

Frame Mixup vs. Video Mixup. We incorporated mixup augmentation to enhance the robustness of our tracking module. Since we adapted this method for video frames, intuitively, one might expect to synthesize intermediate frames by creating a weighted average of two frames and their corresponding heat maps. This strategy, which we refer to as frame mixup, can mimic the motion blur often encountered in broadcast badminton videos. Contrary to this approach, we experimented with another, less intuitive strategy: randomly selecting two video sequences along with their corresponding heat maps and applying the mixup method to them, referred to as video mixup. When we compared these two strategies, the results were unexpected (see Table 3). Frame mixup did not provide a significant improvement to tracking accuracy. However, video mixup did enhance tracking accuracy, particularly when used in conjunction with the estimated background image concatenated to the video frames. This result suggests that more complex examples for augmentation might be necessary when dealing with dynamic video sequences as opposed to individual frames.

Trajectory Rectification. We employ inpainting for missed shuttlecock detections, capitalizing on the shuttlecock's temporal dynamics. Although using a simple linear interpolation of the detected shuttlecock positions from the nearest frames seems plausible, this approach easily fails since shuttlecock trajectories are not always convex, as illustrated in the first two rows of Figure 4. Regarding parameter setting when training the rectification module, we set the mask ratio at 0.3 and the length of input trajectories for rectification at 16 frames. These settings were chosen due to their superior performance experimentally, as detailed in Table 4.

4.6 Limitations

Despite TrackNetV3 demonstrating superior performance over baseline methods in tracking shuttlecocks within broadcast badminton videos, there are still areas for improvement. The system typically encounters failures when shuttlecocks are barely visible, requiring tracking based solely on temporal dynamics. Additionally, the tracked shuttlecock positions are within the video coordinate system and need to be translated into the court coordinate system for deeper analysis.

5 CONCLUSIONS AND FUTURE WORKS

We have implemented a number of strategies to augment shuttlecock tracking accuracy, incorporating estimated background imagery, mixup data augmentation, and trajectory rectification. Experimental outcomes indicate the efficacy of these strategies in enhancing tracking performance. Given that these tracking results could contribute significantly to downstream tactical analyses, we plan to make our implementation publicly accessible. Currently, the shuttlecock position tracking is two-dimensional, which may lead to a skewed analysis of the shuttlecock's height and speed due to perspective projection. Therefore, we plan to explore the reconstruction of 3D shuttlecock trajectories in the future.

ACKNOWLEDGMENTS

We thank the reviewers for their insightful comments. This work is partially supported by the National Science and Technology Council (NSTC), Taiwan, under Contract: 112-2425-H-A49 -001 - and 111-2221-E-A49 -129 -MY3, and by Industrial Technology Research Institute (ITRI), Taiwan.

REFERENCES

[1] Ryan Beal, Georgios Chalkiadakis, Timothy J Norman, and Sarvapali D Ramchurn. 2020. Optimising game tactics for football. *arXiv preprint arXiv:2003.10294* (2020).

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. 2016. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision Workshops*. 850–865.

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).

[4] Wei-Ting Chen, Hsiang-Yun Wu, Yun-An Shih, Chih-Chuan Wang, and Yu-Shuen Wang. 2023. Exploration of Player Behaviours from Broadcast Badminton Videos. In *Computer Graphics Forum*.

[5] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. 2020. Siamese box adaptive network for visual tracking. In *IEEE/CVF conference on computer vision and pattern recognition*. 6668–6677.

[6] Davide Chicco. 2021. Siamese neural networks: An overview. *Artificial neural networks* (2021), 73–94.

[7] Tom Decroos, Jan Van Haaren, and Jesse Davis. 2018. Automatic discovery of tactics in spatio-temporal soccer match data. In *ACM SIGKDD international conference on knowledge discovery & data mining*. 223–232.

[8] Ross Girshick. 2015. Fast R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE conference on computer vision and pattern recognition*. 580–587.

[10] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*. 249–256.

[11] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. 2017. Learning dynamic siamese network for visual object tracking. In *IEEE International conference on computer vision*. 1763–1771.

[12] Yu-Chuan Huang, I-No Liao, Ching-Hsuan Chen, Tsì-Uí İk, and Wen-Chih Peng. 2019. Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 1–8.

[13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[14] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. 2019. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *IEEE/CVF conference on computer vision and pattern recognition*. 4282–4291.

[15] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. 2018. High performance visual tracking with siamese region proposal network. In *IEEE conference on computer vision and pattern recognition*. 8971–8980.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *IEEE international conference on computer vision*. 2980–2988.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *IEEE conference on computer vision and pattern recognition*. 3431–3440.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *IEEE conference on computer vision and pattern recognition*. 779–788.

[19] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: better, faster, stronger. In *IEEE conference on computer vision and pattern recognition*. 7263–7271.

[20] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*. 234–241.

[23] Denys Rozumnyi, Jan Kotera, Filip Sroubek, Lukas Novotny, and Jiri Matas. 2017. The world of fast moving objects. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5203–5211.

[24] Manoj Sharma, Naresh Kumar, Pardeep Kumar, et al. 2021. Badminton match outcome prediction model using Naïve Bayes and Feature Weighting technique. *Journal of Ambient Intelligence and Humanized Computing* 12, 8 (2021), 8441–8455.

[25] Nien-En Sun, Yu-Ching Lin, Shao-Ping Chuang, Tzu-Han Hsu, Dung-Ru Yu, Ho-Yi Chung, and Tsì-Uí İk. 2020. TrackNetV2: Efficient shuttlecock tracking network. In *International Conference on Pervasive Artificial Intelligence (ICPAI)*. 86–91.

[26] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7464–7475.

[27] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. 2018. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *IEEE conference on computer vision and pattern recognition*. 4854–4863.

[28] Wei-Yao Wang, Hong-Han Shuai, Kai-Shiang Chang, and Wen-Chih Peng. 2021. ShuttleNet: Position-aware Fusion of Rally Progress and Player Styles for Stroke Forecasting in Badminton. *arXiv preprint arXiv:2112.01044* (2021).

[29] Tianxiao Zhang, Xiaohan Zhang, Yiju Yang, Zongbo Wang, and Guanghui Wang. 2020. Efficient golf ball detection and tracking based on convolutional neural networks and kalman filter. *arXiv preprint arXiv:2012.09393* (2020).

[30] Zhipeng Zhang and Houwen Peng. 2019. Deeper and wider siamese networks for real-time visual tracking. In *IEEE/CVF conference on computer vision and pattern recognition*. 4591–4600.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009