Trajectory-based Badminton Shots Detection

Nyan-Ping Ju   Dung-Ru Yu   Tsì-Uí İk*   Wen-Chih Peng
Department of Computer Science, College of Computer Science
National Chiao Tung University
1001 University Road, Hsinchu City 30010, Taiwan
*Email: cwyi@nctu.edu.tw

***Abstract*—Shot-by-shot match video segmentation is essential in video-based microscopic data annotation and collection for strategic analysis. With the help of deep learning vision technology, the shuttlecock trajectory can be depicted from broadcast video with accuracy around 78%. In this work, to develop automatic badminton match video labeling, we applied Artificial Neural Networks (ANNs) in the contest strategy data collection to speed up the labeling procedure. The proposed ANN was trained to detect badminton shot events based on shuttlecock trajectories in the contest video. Badminton shot events include serving, hitting, and dead ball. With the help of these shot events, the strategy analyst could annotate strategy information more efficiently and reduce labor costs significantly.**

***Index Terms*—Badminton, shuttlecock, TrackNet, trajectory smoothing, polynomial curve fitting, shots detection, multi-classification, Artificial-Neural Network**

I. Introduction

Badminton is one of the most popular sports over the world. In a tight match, the use of tactics will be the key to success. Traditional tactical analysis is more like worker wisdom, and the quality of outcomes are heavily dependent on the experience of the analyst and sometimes even on luck. However, statistical methods made possible by the availability of big data have gradually become mainstream. Taking badminton as an example, the microscopic level data that may include hitting times of shots, positions, gestures, and/or postures of players, and shot types are indispensable in future precise athlete studies and tactical and strategical analysis. Among these attributes, the hitting time is the key in developing the capability of shot-by-shot microscopic data collection. In this work, based on shuttlecock trajectories computed by deep learning networks, we develop machine learning algorithms to provide shot-by-shot trajectory segmentation and classify the shot events into serving, hitting, and dead ball.

Machine learning algorithms were developed for smart rackets [1] [2] [3] to detect hitting events and classify stroke types from inertial and acoustic sensor data. However, the installation of sensors on the racket may influence player's performance and is not permitted in official games. Computer vision based solutions are another option. Tennis ball and shuttlecock tracking can be categorized as a tiny and high-speed object detection problem. Classical computer vision methods [4] and deep learning networks [5] were proposed to compute ball trajectory from broadcast video. Rule-based heuristics based on trajectory features such as displacement vectors and curvatures were developed to detect shot events. The accuracy of rule-based approaches are usually suffered by changes in the perspective of cameras. Besides, YOLO [6] was applied to detect player's skeleton and classifiers were developed to recognize player's postures. Therefore, strike action can be identified.

In this work, algorithms to detect shot events based on shuttlecock trajectories were developed. The proposed solution can be divided into two parts, the trajectory smoothing module, and the shot detection module. First of all, TrackNet is applied to compute 2D shuttlecock trajectories from the video. Then, since false predictions of TrackNet may seriously affect the accuracy of shot detection, false positives are removed by displacement-based rules, and miss detection and positioning inaccuracy are handled by polynomial curve fitting based smoothing and interpolation. Finally, Artificial Neural Networks (ANN) is designed and trained to detect and classify shot events based on the shuttlecock trajectories. For the purpose of rally segmentation, the shot events are classified into three classes, including serve, return, and dead ball. To make up possible misdetection and false classification of the ANN, the rule-based algorithms are introduced behind the ANN. Based on the time markers, efficient replay systems can be developed. The hitting time of shot events is the most important attribute. It is the indicator to segment match video, snapshot play gestures and postures, and classify ball types no matter in manually or automatic labeling.

The match video of 2018 All England Open Final Rounds Tai Tzu Ying vs Akane Yamaguchi is used in this work for performance evaluation. 70% of the dataset was used in the training of the ANN multi-classifier, and 30% of the dataset was used for evaluation. The precision rate and the recall rate of the ANN multi-classifier are 80% and 79% respectively. Most of the misdetections were due to problematic shuttlecock trajectories that can not be made up by the trajectory smoothing algorithm. How to improve the correctness of trajectories would be the first priority in the future.

The contributions of this paper are two folds. First, to detect and classify badminton shot events based on 2D shuttle trajectories, an ANN model is developed. Based on shot events predicted by an ANN, the tactics data collection process can be more efficient. However, the shuttlecock trajectory accuracy would dramatically affect the performance of the ANN. Which leads to the second contribution, to improve the performance of the proposed ANN model, the shuttlecock trajectories depicted by TrackNet are made up by a polynomial curve fitting

978-1-6654-0483-9/20/$31.00 ©2020 IEEE
DOI 10.1109/ICPAI51961.2020.00020

based smoothing algorithm. The precision rate and recall rate of the following ANN model are significantly improved. The rest of this paper is organized as below. In Section II, background knowledge of this work is introduced. In Section III, the proposed algorithms including the trajectory smoothing module and the shot detection module are given. In Section IV, the performance of the proposed trajectory smoothing algorithm and the ANN multi-classifier for shots detection is discussed. Finally, conclusion remarks and possible future works are given in Section V.

II. Preliminary

A. Related Work

Strategy information play an important role in the high-level badminton contest. Which include monitoring of ball type usage and ball accuracy, evaluating players' reaction agility and footwork techniques, alerting happening of continuous loss and analyzing the causes, and detecting modes of defensive or offensive playing. To provide microscopic level data, all the information above need to be annotated in shot-by-shot level. Therefore, the supporting system to collect microscopic competition data is very important for strategy information collection process.

Smart racket has been used for shot events detection in the badminton contest. Smart racket is designed to recognize stroke types by utilizing wearable IMU sensors installed at the bottom of the racket handle. Wearable IMU sensors have been used for stroke recognition in tennis match. [7] The sensors will transmit information back to the computer and mobile phone via Bluetooth. According to this information, we can accurately detect the hitting event. Also, acoustic sensors could be applied to detect the hitting event. [8] An acoustic shock sensor was mounted on the racket head to identify the instant of the contact event. However, we cannot limit badminton players to use our badminton rackets. Also, the racket equipped with sensors might make players feel slightly different about the racket and it may affect players' performance during badminton contests. Overall speaking, smart racket could be applied to entertainment or practice purposes instead of competition data collection.

Based on computer vision, contest videos can be considered as logs of visual sensors and retain a large amount of information. Shuttlecock trajectory computed by TrackNet could be used to detect shot events based on rule-based heuristics. Trajectory features such as displacement vectors and curvatures were developed to identify shot events. However, the accuracy of the rule-based approach strongly affected by the perspective of cameras. The slight difference of the perspective of cameras might cause trajectory features changed, which would reduce the accuracy of detection. YOLO was applied to detect the player's skeleton and classifiers were developed to recognize player's postures. According to hitting postures, the hitting event could be identified in the contest video. However, the player's postures during running often cause misjudgment. Moreover, slightly hitting postures are not easy to detect by YOLO.

B. Strategy Analysis

In the high-rank badminton contest, the outstanding performance of players depends on not only solid training but also tactical information during matches. Based on the microscopic competition data. the strategy analysis is going to build the capability of strategy analysis and recommendation based on machine learning techniques and big data analysis. The tactical information not only will be provided for athletes and coaches' reference to boost the training efficiency. Statistic analysis can be applied to analyze the footwork patterns and to find the reasons of continuous loss. Spatial analysis can help to understand the ability of ball control and to find ball usage patterns. Temporal analysis can be used to estimate physical declination and to detect tactical change. During the matches, the capability of real-time tactical analysis is important in order to provide tactical information for coaches and players

C. TrackNet

A CNN based framework, called TrackNet, has been proposed to detect small objects such as tennis. TrackNet is designed to process three consecutive frames at once and output a ball detection heatmap of the last frame. Since it calculated three frames at once, not only the appearance but also the trajectory of badminton could be processed for ball detection. In this work, we adopted TrackNet to track the high speed shuttlecock from broadcast videos or those token by consumer mobile devices such as smartphones. The framework of TrackNet as shown in Figure 1.

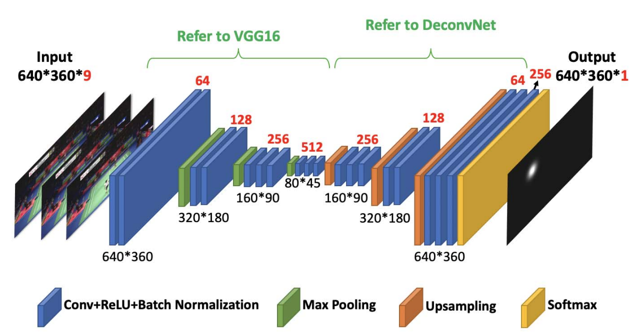

Fig. 1: The architecture of TrackNet.

D. Polynomial Curve Fitting

The curve fitting function could quickly and easily fit a set of data points with a polynomial equation. Polyfit is a function that computes the least squares polynomial for a given set of data. Polyfit generates the coefficients of the polynomial, which can be used to model a curve to fit the data. Curve fitting [9] involves either interpolation which function exactly fits the data, or smoothing which function is constructed that approximately fits the data. Fitted curves could be used to predict the values of a function where the data are missing. Also, extrapolation refers to the use of a fitted curve beyond the range of the observed data which could predict the missing data with a degree of uncertainty. In this work, the fitted curve of badminton trajectories was computed by polyfit function in Numpy. [10] Based on the fitted curve, the misjudgment and the missing coordinates of TrackNet could be modified.

E. Artificial Neural Network multi-classifier

Artificial Neural Networks(ANN) are multi-layer fully connected neural networks. It consists of an input layer, multiple hidden layers, and an output layer. ANN has been used on a variety of tasks, including computer vision, speech recognition, and even medical diagnosis. ANN is also effective in categorizing data into identifiable groups. The most famous example is applying ANN in iris recognition [11], based on iris unique features ANN multi-classifier can achieve high classification accuracy. In this work, the ANN multi-classifier was used to classify shot events according to badminton trajectory features. With the help of the ANN multi-classifier, the efficiency of the strategy information collection process could be improved and reduce labor costs significantly.

III. Trajectory Smoothing and Shots Detection Algorithms

The main purpose of this paper is to use ANN to improve the badminton contest strategy data collection process. In order to reach this goal, we used the convolution neural network, TrackNet, to predict the badminton coordinates of each frame in the contest video. By these coordinates and features, we trained an ANN to predict shot conditions of each frame. As shown in Figure 2, the pipeline can be separated into two sections to discuss. First, shuttlecock trajectory module. In this section, we will discuss the algorithm we used to smooth and fill the badminton coordinates. Second, shots detection module. The person who is labeling the contest video used to review video manually to annotate strategy data of these special shot conditions. In this section, we will discuss the training features, the architecture of the ANN multi-classifier and the rule-based algorithm which were used to improve the accuracy of this ANN multi-classifier.

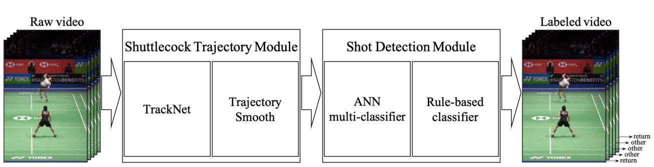

Fig. 2: Pipeline.

A. Shuttlecock Trajectory Module

The accuracy of shuttlecock detection is the major factor in the construction of badminton trajectory and directly affects the performance of shots detection. However, due to the high moving speed, the badminton images in the video may be blur and cannot be detected by TrackNet. According to the statistics, the shuttlecock trajectory can be depicted from broadcast video with an accuracy of around 80%. Therefore, in this paper, we proposed the trajectory smoothing algorithm to improve the accuracy of TrackNet. TrackNet suffers from two major problems which may affect the performance of shots detection. Figure 3(a) refers to the normal trajectory, the trajectory should approximate to the curve of second order and coordinates of each frame should exist. However, video conditions like badminton images are blur or background noise which looks like badminton will make TrackNet misjudge accurate coordinates in the video. When the Euclidean distance of predicted coordinates and actual coordinates is larger than 10 pixels, we defined this situation as shifted coordinate, as shown in Figure 3(b). Another situation like coordinates are empty because of TrackNet can not locate badminton in the video, we defined this situation as missing coordinates, as shown in Figure 3(c).

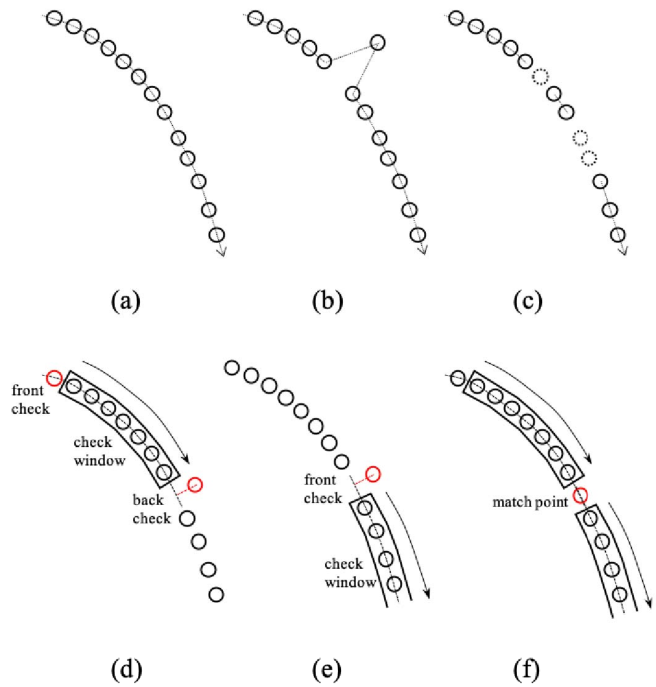

Fig. 3: Example of a trajectory problems.

In order to solve these problems, the proposed algorithm can be separated into three phases to discuss. The first phase is denoise. In this phase, we want to eliminate shifted coordinates as much as possible and leave the accurate coordinates for the next phase. According to the statistics, around 9% of predicted coordinates shifted more than 100 pixels and 2.4% of predicted coordinates shifted between 10 pixels and 100 pixels. Therefore, in this phase, we used the Euclidean distance between coordinates of each frame to remove the predicted coordinates which shifted more than 100 pixels. When the distance between coordinates of the current frame and previous frame is larger than 100 pixels, also the distance between coordinates of the current frame and next frame is larger than 100 pixels, we defined the coordinate of the current frame as shifted coordinate. According to this algorithm, we roughly eliminated around 9% of predicted coordinates which shifted more than 100 pixels. Predicted coordinates shifted between

10 pixels and 100 pixels can not be removed because of the distance between abnormal coordinates is no larger than the distance between normal coordinates. Thus, we will deal with it in the next phase.

The second phase of this algorithm is curve fitting. In this phase, we used quadratic equations which composed of coordinates from the first phase to smooth and fill the trajectory. As shown in Figure 3(d), We define seven consecutive sets of coordinates as check window. If there existed at least three coordinates in this check window, we can calculate quadratic equation of these coordinates. According to this quadratic equation, we defined the minimum distance from coordinate before the check window to the curve as front check distance. Similarly, the minimum distance from coordinate after the check window to the curve as back check distance. For each coordinate, it has the back check distance of the previous check window and the front check distance of the following check window. If the back check distance and the front check distance are abnormally large, it means that the current coordinate deviates the trajectory which composed of 1check windows nearby it. Therefore, this coordinate will be defined as shifted coordinate, as shown in Figure 3(e). With the help of this algorithm, we can remove shifted coordinates which phase one can not detect. On the other hand, if the back check distance and the front check distance are smaller than 5 pixels, it means that current coordinates fit the trajectory which composed of check windows nearby it. Consequently, we defined this coordinate as a match point, as shown in Figure 3(f). In this phase, we used match points as the filling standard. When the match point happened and there are missing coordinates or shifted coordinates in the check windows nearby it, we can use the quadratic equation of each check windows to fill coordinates.

After the second phase, most of the coordinates had been adjusted and the trajectory becomes more complete. However, there are still some missing coordinates left behind. Thus, in the third phase, we used interpolation to fill the rest of the missing coordinates. When there are three coordinates before and after missing coordinates, we used these coordinates to calculate the quadratic equation and fill missing coordinates. However, there is a limitation of this algorithm. If the consecutive missing coordinates greater than five coordinates, filled coordinates would be not accurate enough. The original trajectory of consecutive missing coordinates can be complicated, a situation like players returning the badminton might happen in this missing period. Therefore, in phase three we only fill consecutive missing coordinates not greater than five coordinates

Shuttlecock Trajectory Smoothing Algorithm

```
Phase 1: Denoise.
if distance from current frame to prior frame and
  following frame are more than 100 pixels
 then
  | mark the frame as shifted coordinate;
  | remove shifted coordinate;
end
Phase 2: Curve fitting.
while more than 2 coordinates in check window do
  | calculate quadratic equation
  | if coordinate exists before check window then
  |   | mark minimum distance from coordinate to
  |   |   quadratic equation as front check distance;
  | end
  | if coordinate exists after check window then
  |   | mark minimum distance from coordinate to
  |   |   quadratic equation as back check distance;
  | end
end
if front check and back check distance more than 100
  pixels then
  | mark coordinates as shifted coordinates;
  | remove shifted coordinates;
end
if front check and back check distance more than 5
  pixels then
  | mark coordinate as match point;
  | fill missing coordinates by quadratic equation;
end
Phase 3: Interpolation.
if exists 3 coordinates before and after missing
  coordinates then
  | calculate quadratic equation by these 6 coordinates;
  | fill missing coordinates by quadratic equation;
  | return;
else
  | return;
end
```

B. *Shots Detection Module*

In this section, we used information from the shuttlecock trajectory module to build an ANN multi-classifier which can detect shot conditions of each frame. The badminton trajectory is a continuous movement. In order to show its continuity, features of each frame were calculated including coordinates before and after the current frame. As shown in Figure 4, when the frame window total includes 9 frames, the precision rate and the recall rate can reach to almost 99%. The frame window includes 11 frames shows no further improvement for precision and recall rate. Therefore, in our experiment, we used the frame window total includes 9 frames to calculate features for the ANN multi-classifier. Features include coordinates of each frame in the frame window, the distance between each frame, vectors between each frame, vector difference, vector multiplication, vector dot product and angles between each frame. Prediction targets of these features include shot conditions like rally start, each player returns the badminton, rally end and other. The shot condition of rally start means the timing when the player served the badminton. After that, when each player returned the badminton, the shot condition will be defined as the return. Finally, when the badminton fell to the ground, the shot condition will be defined as the rally end. The shot condition of others includes when the badminton was flying and the badminton was staying on the ground. However, most of the shot condition of these features is other, conditions like rally start, return and rally end are relatively rare. Thus, before trained the ANN multi-classifier, we used random undersampling [12] to re-sample the shot condition like others and made it balanced with shot conditions like rally start, return and rally end.

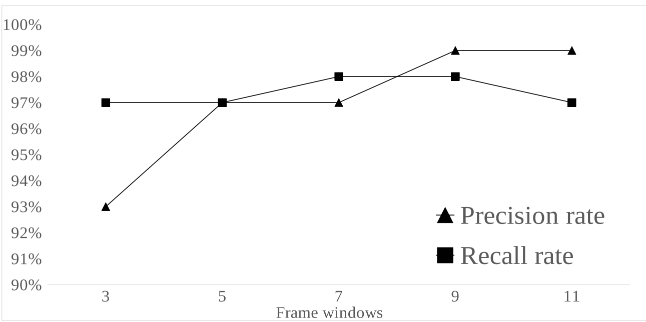

Fig. 4: Frame window line chart.

As shown in Table I, the ANN multi-classifier is composed of 6 layers. The first layer is the batch normalize layer. Because of the various scale of training features might affect the ANN multi-classifier performance. We used it to standardize the input dataset of the next hidden layer. This has the effect of stabilizing the learning process and improve the training efficiency of ANN. [13] After the batch normalize layer, there are two sets of hidden layers and dropout layers. For the hidden layer, in order to prevent the gradient descent, we used ReLU as an activation function. The kernel initializer is random normal and the number of nodes is 100. In order to adapt different shot angles of contest videos, we used the dropout layer to prevent overfitting in neural networks. [14] In the end, we used softmax as the activation function of the final layer. The output of this layer is the probability of each shot conditions, the shot condition with the highest probability stand for the representative shot condition of that frame. During training, we used Adam as our optimizer, and categorical cross-entropy as our loss function. The batch size of ANN is 32, and the training epochs is 100.

TABLE I: ANN Multi-classifier Architecture

Layer	Node	Activation	Kernel initializer	Input dimension
1	Batch normalize layer			
2	100	RuLU	Random normal	89
3	Dropout layer			
4	100	ReLU	Random normal	-
5	Dropout layer			
6	4	Softmax	Random normal	-

However, classified shot conditions of each frame are not the final result. After the ANN multi-classifier, we used the rule-based algorithm to enhance classification accuracy. The first phase of this algorithm is voting. Most of the time, when badminton trajectory dramatically changes it means special shot conditions happened, such as rally start, return and rally end. However, trajectory change would not just happen in one frame, it is a continuous movement and makes the ANN classified not only one special condition during that period. In order to prevent this situation, we used voting to choose the special condition which has the highest score within 6 frames as a representative shot condition of that period. The second phase and third phase of this algorithm are similar, the rally start classification and the rally end classification. Because of the shot conditions like rally start and rally end is fewer than return during the badminton contest, there is a high probability that ANN misclassified shot conditions of rally start and rally end into return. Therefore, we used these algorithms to reclassify the result. During badminton contest, most of the players will raise the badminton in front of them and keep steady as a preparation movement before serving the badminton. Which leads to the second phase, when special shot conditions happened, we will check 30 frames prior to this shot condition. If there are 5 continuous frames satisfied Euclidean distance between the frame nearby less than 5 pixels, it means that the player prepared for serving before this special shot condition. Thus, we replaced this shot condition with rally start. Similarly, after badminton landing the ground, there is a period of time that badminton keeps steady on the ground until players pick it up. This leads to the third phase, when the special shot condition happened, we will check 30 frames after this shot condition. If there are 5 continuous frames satisfied Euclidean distance between the frame nearby less than 5 pixels, it means that the badminton keeps steady on the ground after this shot condition happened. Therefore, we replaced this shot condition with rally end.

IV. Performance Evaluation

A. Dataset

In this paper, we used the match of Tai Tzu-Ying vs. Akane Yamaguchi in 2018 All England Open as our dataset. There are about 15,000 frames in this contest video. 70% of the dataset was used for training and 30% was used for testing. We used badminton coordinates from three dataset sources to evaluate trajectory smoothing and shots detection performance: 1)Manual label 2) TrackNet predicted result 3)TrackNet predicted result with shuttlecock trajectory smoothing.

B. Evaluation of trajectory smoothing

In order to detect shot conditions based on badminton trajectory, we need to provide coordinates as accurately as possible for ANN multi-classifier. Thus, in this paper, we proposed the trajectory smoothing algorithm to enhanced the trajectory accuracy of TrackNet. In this section, we evaluated trajectory smoothing algorithm performance based on three

Rule-based Classification Algorithm

Phase 1: Voting.
labeled special shot conditions of each frame;
if *special shot conditions exists in 6 frames* **then**
  keep the shot condition with the highest score;
  replace rest of shot conditions as other;
  goto Phase 2;
else
  goto Phase 2;
end

Phase 2: Rally start classification.
calculate Euclidean distance between each frames;
if *special shot condition exists* **then**
  if *more than 5 continuous frames Euclidean distance lower than 5 pixels in 30 frames before this frame* **then**
    mark shot condition of this frame as rally start;
  else
    goto Phase3;
  end
else
  goto Phase3;
end

Phase 3: Rally end classification.
calculate Euclidean distance between each frames;
if *special shot condition exists* **then**
  if *more than 5 continuous frames Euclidean distance lower than 5 pixels in 30 frames after this frame* **then**
    mark shot condition of this frame as rally end;
  else
    return;
  end
else
  return;
end

dataset sources. The most precise dataset source is badminton coordinates from manual labeling, badminton coordinates exist in each frame other than conditions like badminton occluded by players or badminton flew out of the screen. Therefore, we used coordinates from manual labeling as our standard to evaluate the other two datasets. To evaluate the accuracy of datasets, for 1280x720 resolution video, we took 10 pixels Euclidean distance as our offset between manually labeled coordinates and corresponded coordinates of other datasets. As shown in Table II, the accuracy of TrackNet within 10 pixels offset can achieve to around 78%. After processed by trajectory smoothing algorithm, the accuracy can enhance to 83% when filled coordinates within 10 pixels offset. However, the distance between manually labeled coordinates and filled coordinates was mostly around 10 pixels to 50 pixels. Therefore, if we do not consider 10 pixels as offset for filled coordinates, the accuracy of TrackNet with trajectory smoothing can reach to 92%. Although the filled trajectory may not as accurate as original trajectory, it can still represent the turning characteristic and detected by the ANN multi-classifier. The shot detection performance of these datasets will show in the next section.

TABLE II: Trajectory Smoothing Accuracy

	Manual label	TrackNet (w/ offset)	TrackNet w/ smoothing (w/ offset)	TrackNet w/ smoothing (w/o offset)
Coordinates	3983	3116	3289	3672
Accuracy	100%	78%	83%	92%

C. Evaluation of shots detection

In this section, we used 30% of the contest video as our testing dataset to evaluate shot detection performance. Testing features were calculated by three different dataset sources from previous sections. The outputs of the ANN multi-classifier were four kinds of shots condition, such as rally start, return, rally end and other. The ground truth of the shot condition was manually labeled for each frame. However, the predicted result may not exactly match with the ground truth. Although the predicted shot condition is the same as ground truth, they might not exactly correspond to the same frame. Therefore, when we evaluated the ANN multi-classifier, we took 5 frames before and after the ground truth shot condition as our offset. When the predicted shot condition was the same as ground truth and located nearby ground truth within 5 frames, the predicted shot condition would be regarded as true positive. In this situation, the predicted shot condition was only shifted about 0.2 seconds from the manual labeled shot condition for the 30 fps contest video, which is tolerable for the strategy information collection process.

In this experiment, our major goal is to use ANN to classify special shot conditions like rally start, return and rally end. Thus, we count these special shot conditions as true positive, and other as true negative. Because of the imbalanced dataset problems, we used precision rate and recall rate as our key performance indicators to evaluate shot detection performance. The equation of precision rate and recall rate as follows:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

The shot detection performance can be separated into two situations to discuss. In the first situation, we want to separate shot conditions into four types, such as rally start, return, rally end and other. According to these shot conditions, we can separate the contest video into rallies and label match scores of it. On the other hand, for strategy information collection purpose, we only need the ANN to classify two types of shot conditions, such as trajectory turn and other. In this situation, the shot condition of trajectory turn includes all the special shot conditions in the previous situation.

The confusion matrix of the ANN multi-classifier based on three dataset sources were shown in Table III. Based on manually labeled coordinates, the trajectory was closest to the real trajectory in the contest video. The precision rate of ANN can reach to 99% and the recall rate can reach to 97%. For the false negative cases, three frames of return were misclassified into other because the trajectory didn't change dramatically during player return the badminton, and only one frame of return was truly missed by the ANN. For the false positive case, one frame of other was misclassified into rally end because after badminton fell to the ground, it bounced back and fell again. Therefore, the ANN detected two rally end at that time and marked the second rally end as false positive. Based on TrackNet coordinates, the precision rate of ANN was 70% and the recall rate was 31%. The recall rate drop dramatically because of the trajectory accuracy of TrackNet was only 78%, and incomplete features lead to the poor classification of ANN multi-classifier. To prevent this situation, we used the trajectory smoothing algorithm to adjust and fill TrackNet coordinates. Based on this dataset, the precision rate can reach to 80% and recall rate can reach to 79%. When we only considered two types of shot conditions, the precision rate can reach to 85% and the recall rate can reach to 82%. According to this result, we can conclude that trajectory accuracy can dramatically affect the performance of shots detection and the trajectory smoothing algorithm can make up for the deficiency of TrackNet coordinates. However, there's still room for improvement. The false positive and false negative cases were because of the false coordinates which trajectory smoothing algorithm could not modify. The future works will focus on the method to improve the accuracy of TrackNet and further algorithms to adjust and fill coordinates of TrackNet.

TABLE III: ANN Multi-classifier Confusion Matrix

Manual label coordinates

actual\predict		start	return	end	other
start		2	0	0	0
return		0	115	0	4
end		0	0	10	0
other		0	0	1	4532
Four conditions (start/return/end/other)	Precision	99%			
	Recall	97%			
Two conditions (turn/other)	Precision	99%			
	Recall	97%			

TrackNet coordinates

actual\predict		start	return	end	other
start		0	0	0	2
return		0	38	1	80
end		0	0	2	8
other		1	15	1	4516
Four conditions (start/return/end/other)	Precision	70%			
	Recall	31%			
Two conditions (turn/other)	Precision	71%			
	Recall	31%			

TrackNet with trajectory smoothing coordinates

actual\predict		start	return	end	other
start		1	0	0	1
return		4	93	3	19
end		1	0	5	4
other		4	11	4	4514
Four conditions (start/return/end/other)	Precision	80%			
	Recall	79%			
Two conditions (turn/other)	Precision	85%			
	Recall	82%			

V. Conclusion

In this paper, in order to improve the efficiency of the badminton contest strategy information collection process, we utilized ANN multi-classifier with a trajectory smoothing algorithm to speed up the labeling procedure. The precision rate and recall rate can reach to 80% and 79% for four types of shot conditions such as rally start, return, rally end and other; 85% and 82% for two types of shot conditions such as trajectory turn and other. The lost precision rate and recall rate was major because of the trajectory accuracy is still not good enough. Future work will focus on how to further improve the accuracy of TrackNet and algorithm to modify the false coordinate of TrackNet.

Acknowledgement

This work of T.-U. İk was supported in part by the Ministry of Science and Technology (MOST), Taiwan under grant MOST 109-2627-H-009-001. This work was financially supported by the Center for Open Intelligent Connectivity from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

References

[1] T.-H. Hsu, C.-H. Chen, N. P. Jut, T.-U. İk, W.-C. Peng, Y.-S. Wang, Y.-C. Tsengt, J.-L. Huang, Y.-T. Ching, C.-C. Wang *et al.*, "Coachai: A project for microscopic badminton match data collection and tactical analysis," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2019, pp. 1–4.

[2] J. Lin, C.-W. Chang, C.-H. Wang, H.-C. Chi, C.-W. Yi, Y.-C. Tseng, and C.-C. Wang, "Design and implement a mobile badminton stroke classification system," in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2017, pp. 235–238.

[3] Y.-C. Huang, T.-L. Chen, B.-C. Chiu, C.-W. Yi, C.-W. Lin, Y.-J. Yeh, and L.-C. Kuo, "Calculate golf swing trajectories from imu sensing data," in *2012 41st International Conference on Parallel Processing Workshops*. IEEE, 2012, pp. 505–513.

[4] G. Waghmare, S. Borkar, V. Saley, H. Chinchore, and S. Wabale, "Badminton shuttlecock detection and prediction of trajectory using multiple 2 dimensional scanners," in *2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*. IEEE, 2016, pp. 234–238.

[5] Y.-C. Huang, I.-N. Liao, C.-H. Chen, T.-U. İk, and W.-C. Peng, "Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications," in *The 1st IEEE International Workshop of Content-Aware Video Analysis (CAVA 2019) in conjunction with the 16th IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS 2019)*, 18-21 September 2019.

[6] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[7] M. Kos, J. Ženko, D. Vlaj, and I. Kramberger, "Tennis stroke detection and classification using miniature wearable imu device," in *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2016, pp. 1–4.

[8] C. T. Kiang, C. K. Yoong, and A. Spowage, "Local sensor system for badminton smash analysis," in *2009 IEEE Instrumentation and Measurement Technology Conference*. IEEE, 2009, pp. 883–888.
[9] P. G. Guest and P. G. Guest, *Numerical methods of curve fitting*. Cambridge University Press, 2012.
[10] W. M. Kolb, *Curve fitting for programmable calculators*. Imtec, 1984.
[11] F. N. Sibai, H. I. Hosani, R. M. Naqbi, S. Dhanhani, and S. Shehhi, "Iris recognition using artificial neural networks," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5940–5946, 2011.
[12] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.
[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.