Recognizing and Predicting Opponent's Badminton Shot Types*

1st Chang, Tzu-Hung
Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
b09902050@csie.ntu.edu.tw

2nd Lee, Yuan-Chi
Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
b09902110@csie.ntu.edu.tw

3rd Chou, Hung-Yi
Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
b09902113@csie.ntu.edu.tw

4th Tu, Yu-Chieh
Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
b09902138@csie.ntu.edu.tw

***Abstract*—Badminton is a popular sport characterized by its fast-paced nature and diverse shot types. Recognizing and predicting the shot type in real-time can provide valuable insights for coaches, players, and spectators. In our work, first, we define a new task to recognize and predict shot types. Then, we present a methodology to try enhancing shot type prediction in badminton videos by adding information to the video frames.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

For the task, the most natural way to predict the shot types is to guess from the movement of the opponent. However, in the professional-level badminton tournaments, we can observe that the shot types have a high degree of concealment, in terms of the player's movement. For instance, one can hardly distinguish between clear shot, drop shot, and smash shot before the shuttlecock is actually hit. Then, except for the movement, we claim more patterns may emerge from the positions of both players and the trajectory of shuttlecock within a few seconds. Thus, we leverage two extra models to fetch additional information. More specifically, two type of annotations are added to the videos frames: 1. The bounding boxes of players 2. The trajectory

By doing this, we expect the model to focus more on the highly related information.

II. Dataset

We chose three badminton games on the BWF official channel, and then sampled 1200 clips from these videos. All of these clips are last for 2s, and they contains clips from different angle of views, such as from the top of the court, from the left side or right side of the court. And some of these clips do not contain competition footage which is then classified to the "None" class, we will talk about this later. There are 1000 clips in the training set, and 200 clips in the validation set. We classified these clips into 8 classes, which are "Clear", "Drop", "Drive", "Smash", "Net", "Lift", "Kill", and "None". Each clip corresponds to a category, and we determined the class of every clip by the shot the opponent hit at the last moment of every clip.

What we did in this final project is either to recognize or to predict the shot types hit by the opponent. In the following statement, we regard the player closer to us as "ourselves", and the player farther to us as "the opponent". In the period after we hit the shuttlecock and before the opponent did not hit the shuttlecock, we "make a prediction" about the next shot type made by the opponent. In the period before we hit the shuttlecock and after the opponent hit the shuttlecock, we will "recognize" the shot type hit by the opponent.

Besides the dataset which has no annotations on the clips, we also build two other datasets which have annotations on every clip. One is called "Ball dataset", and the other one is called "Ball Person Dataset". We manage to improve the performance and to help models learn better by the two datasets.

A. Ball Dataset

To provide neural networks with more information and assist in their learning, we annotated additional data in the training set. Firstly, we marked the trajectory of the ball. We utilized a model called TrackNet V2 to identify the path of the ball. The design of TrackNet V2 is as follows: it consists of an encoder-decoder structure. In the encoder, the model takes in three consecutive frames of the video at a time, aiming to capture the fast movement of the ball. Subsequently, the encoder-decoder generates a heatmap, which represents the likelihood of the ball's presence in each pixel of the frame. This heatmap contains information about the ball, and then the model proceeds to locate the ball's position by identifying its contour.

Fig. 1. Dataset with shuttlecock labeled

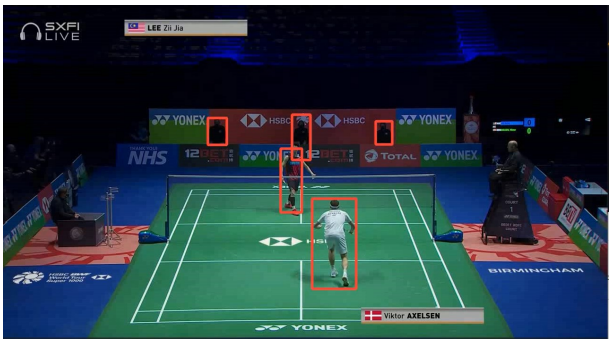

Fig. 2. Ball person dataset before filtering

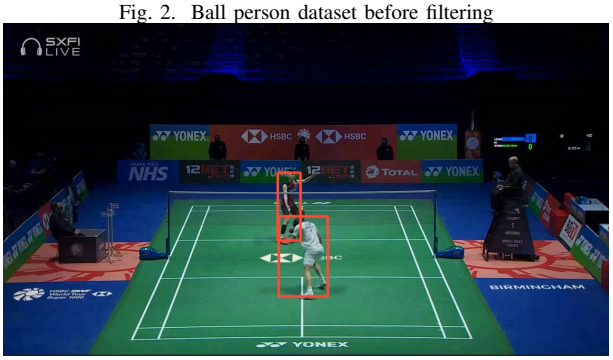

Fig. 3. Ball person dataset after filtering

B. Ball Person Dataset

In addition to tracking the shuttlecock, we also added bounding boxes to the players, hoping to make the model more focused on the players' positions and improve the classification performance. We used YOLOv8 for object detection in this final project. However, since there are other people in the video (such as referees) besides the two players, we cannot directly use the inference results from YOLO.

Therefore, we introduced a rule-based method to filter out the bounding boxes of other people. Before drawing bounding boxes on the frames, we manually identified the court area at the pixel level. Since referees would not enter the court during the game, we can filter out their bounding boxes by only considering the bounding boxes inside the court.

III. Training Method

We use the "slowfast network" introduced by Facebook AI Research. There are two main components, the slow pathway and the fast pathway. The slow pathway mainly aims at extracting the spatial features in a video, and the fast pathway mainly aims at extracting the temporal features in a video. So we sample the input videos with low frame rate but more channels in the slow pathway, and with high frame rate but less channels in the fast pathway. There is a lateral connection from the fast pathway to the slow pathway, according to the original paper, it is fine to either choose summation or concatenation to make the lateral connection. And the final vector would then be fed into the fully connected classifier layer.

A. Experiment

We used the slowfast network to train on the three datasets that mentioned earlier. We set the epochs to 100 epochs and other configs were followed by the yaml released by the official repository. After training for the three datasets, we test the best checkpoint on the validation set and caculated the top1 accuracy, top2 accuracy, and top3 accuracy. The validation set we used to test after training is slightly different with the one used during training, we removed all clips that belong to the "None" class because we thought that the "None" class is much easier to predict, however, it makes up almost one third of the validation set.

B. Training

We trained the slowfast network on three different datasets, the Original dataset, the Ball Dataset, and the Ball Person dataset respectively. During training, the loss curve decreases smoothly, and the validation error(top1-error) also has the decreasing tendency. Among the three datasets, the model decreases the fastest when training on the Ball Dataset and the slowest when training on the Ball Person Dataset.

When training with the Original dataset, the best validation performance happened after 80 epochs of training, reaching an accuracy of 70%. In the Ball Person dataset, the bounding box annotations are less stable, for example, it may appear in some frames and disappear in a few consecutive frames, and then reappear in the following frames. This situation may affect the training, making the model focus on wrong features. With this dataset, the best validation performance happened after 80 epochs of training, reaching an accuracy of 71%. In the Ball dataset, the red annotation of the trace of the shuttlecock makes the training loss decrease faster. With this dataset, the best validation performance happened after 80 epochs of training, reaching an accuracy of 72.5%.

C. Testing

We tested the above models with the validation set, excluding the "None" class, using three metrics: top-1 accuracy, top-2 accuracy, and top-3 accuracy. The completed results can be seen in Table I and Table II.

When using top-1 accuracy as the metric, the model trained on the Original dataset reached the highest accuracy after

30 epochs of training, which is 23.88%. The one trained on Ball dataset reached an accuracy of 16.42% after 70 epochs of training. And the one trained on the Ball Person dataset reached an accuracy of 14.93% after 70 epochs of training. When using top-2 accuracy as the metric, the model trained on the Ball dataset reached the highest accuracy after 50 epochs of training, which is 41.79%. The one trained on Original dataset reached an accuracy of 34.33% after 60 epochs of training. And the one trained on the Ball Person dataset reached an accuracy of 34.% after 40 epochs of training. When using top-3 accuracy as the metric, the model trained on the Ball Person dataset reached the highest accuracy after 50 epochs of training, which is 59.70%. The one trained on Ball dataset reached an accuracy of 58.21% after 20 epochs of training. And the one trained on the Original dataset reached an accuracy of 58.21% after 20 epochs of training.

TABLE I
TOP-n ACCURACY TRAINED ON DIFFERENT DATASET

	Original	Ball Person	Ball
top-1 acc	**23.88**	14.93	16.42
top-2 acc	34.33	34.33	**41.79**
top-3 acc	58.21	**59.70**	58.21

TABLE II
TABLE TITLE

	Original	Ball Person	Ball
training epochs for best top-1	**30**	70	70
training epochs for best top-2	60	40	**50**
training epochs for best top-3	20	**50**	20

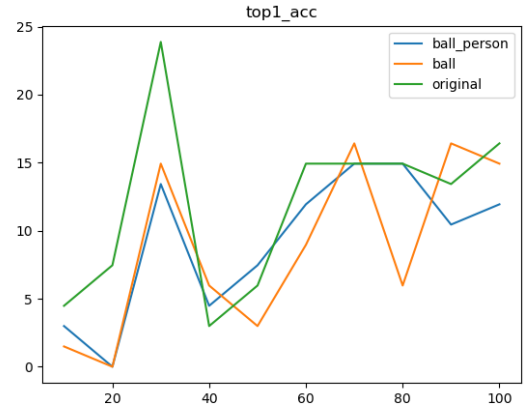

Fig. 4. Top-1 Accuracy Curve

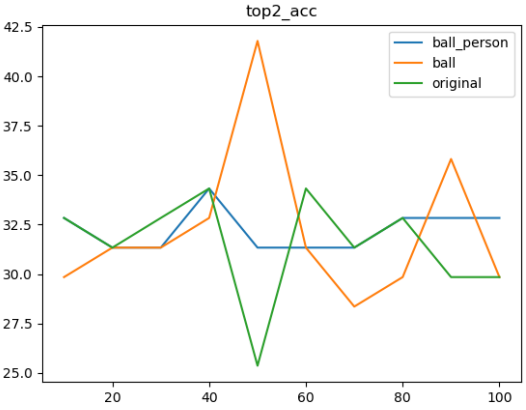

Fig. 5. Top-2 Accuracy Curve

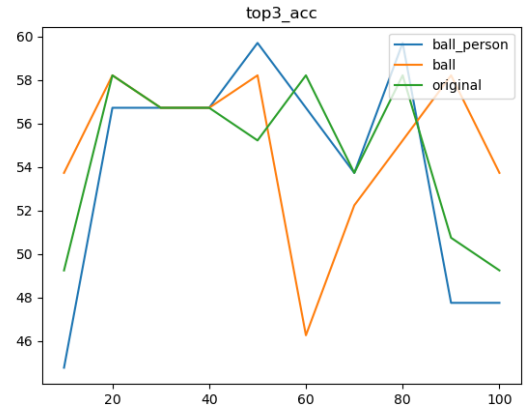

Fig. 6. Top-3 Accuracy Curve

D. Discussion

First, we observed that the training loss curve didn't converge enough, but due to the training time of 100 epochs per day, we considered that we didn't have enough time to make the training epochs longer and train the model in new settings again, so we chose to compare these models by training them for 100 epochs. Secondly, with the limitation of the training time and the dataset, we reach the highest accuracy of 72.5%, meaning that except for about 60% of the none class in the validation set, we still predict the other shot types correct for about 25%, and under the metric of top-1 accuracy, the validation curve is still growing. Maybe we can explore more after we extend the training time. Third, we found that additional annotations on videos may significantly influence the training result, even if the annotations are really within a small area, like the ones in Ball dataset. They still cause the convergent speed of the model becoming the fastest among the three. Finally, we observed that while there is a clear increasing tendency in the validation curve of top-1 accuracy, there is no obvious improvement in top-2 and top-3 accuracy. See Fig. 4, 5, 6. It's probably due to the unbalance of our training data. We thought that the models learn less from classes with limited data, so they tend to avoid predicting these classes, but to predict from those classes with enough amounts of training data. Under this situation though, models can still learn to differentiate among those major classes, and predict the top-1 candidate correctly.

IV. Contribution

First, we build up a dataset for the task of predicting and recognizing the shot types in badminton. Moreover, we made two different types of data transformation on the original dataset to focalize all the crucial parts for classifying videos. Second, we try to analyze the videos and make predictions about the opponent's next move at every moment in the video. Third, we found that the additional annotations on videos have a significant impact on the training process, regardless of how much the area was transformed. Finally, we proposed many possible reasons to those problems we faced during training. Many of them have clear studying directions when given more time.

V. Conclusion

To sum up, our work aimed to enhance the prediction and recognition of shot types in videos of badminton tournaments by incorporating additional information from player bounding boxes and shuttlecock trajectories. We constructed a dataset and implemented two different data transformation techniques to emphasize the critical aspects of video classification. Furthermore, we enable predictions of the opponent's next shot type at any moment. Our experiments demonstrated the significant impact of the added annotations on the training process. While we encountered challenges during training, we proposed several potential explanations and identified promising directions for further investigation. Overall, our work contributes to the advancement of shot type prediction in badminton. With continued research and exploration, we anticipate even greater improvements in this field.

References

[1] 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," arXiv:1907.03698 [cs, stat], Jul. 2019,

[2] N. -E. Sun et al., "TrackNetV2: Efficient Shuttlecock Tracking Network," 2020 International Conference on Pervasive Artificial Intelligence (ICPAI), Taipei, Taiwan, 2020, pp. 86-91, doi: 10.1109/ICPAI51961.2020.00023.

[3] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," GitHub, Jan. 01, 2023. https://github.com/ultralytics/ultralytics

[4] M. Broström, "Real-time multi-object, segmentation and pose tracking using Yolov8 with DeepOCSORT and LightMBN," GitHub, Jun. 11, 2023. https://github.com/mikel-brostrom

[5] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast Networks for Video Recognition," 2019.