

rics (e.g., player’s reaction time, dominance, positioning and footwork around the court, etc.) for higher level analytics. Similar to many other sports, badminton has specific game grammar (turn-based strokes, winning points, etc.), well separated playing areas (courts), structured as a series of events (points, rallies, and winning points), and therefore, are suited well for performing analytics at a very large scale. There are several benefits of such systems. Quantitative scores summarizes player’s performance while qualitative game analysis enriches viewing experience. Player’s strategy, strengths, and weaknesses could be mined and easily highlighted for training. It automates several aspects of analysis traditionally done manually by experts and coaches.

Badminton poses different difficulties for its automatic analysis. The actions (or strokes) are intermittent, fast paced, have complex movements, and sometimes occluded by the other player. Further, the best players employ various subtle deception strategies to fool the human opponent. The task becomes even more difficult with unstructured broadcast videos. The cameras have an oblique or overhead view of players and certain crucial aspects such as wrist and leg movements of both players may not be visible in the same frame. Tracking players across different views makes the problem even more complicated. We discard the highlights and process only clips from *behind the baseline* views which focus on both players and have minimal camera movements. For players detection, we rely on robust deep learning detection techniques. Our frame level stroke recognition module makes use of deep learned discriminative features within each player’s spatio-temporal cuboid.

The major contributions of this paper are

1. We propose an end-to-end framework to automatically annotate badminton broadcast videos. Unlike previous approaches, our method does not rely on special camera setup or additional sensors.
2. Leveraging recent advancements in object detection, action recognition and temporal segmentation, we predict game points and its outcome, players’ tracks as well as their strokes.
3. We identify various understandable metrics, computed using our framework, for match and player analysis as well as qualitative understanding of badminton games.
4. We introduce a large collection of badminton broadcast videos with match level point segments and outcomes as well as frame level players’ tracks and their strokes. We use the official broadcast videos of matches played in London Olympics 2012.

2. Related Work

Sports Understanding and Applications: Several researchers have worked on improving sports understanding using domain specific cues in the past [5, 29]. Racket sports have received a lot of attention in this area with strides made in video summarization and highlight generation [9, 10] and generating text descriptions [36]. Reno et al. [28] proposed a platform for tennis which extract 3D ball trajectories using a specialized camera setup. Yoshikawa et al. [42] performed serve scene detection for badminton games with a specialized overhead camera setup. Zhu et al. [6] performed semi-automatic badminton video analysis by detecting the court and players, classifying strokes and clustering player strategy into offensive or defensive. Mlakar et al. [20] performed shot classification while Bertasius et al. [3] assessed a basketball player’s performance using videos from wearable devices. Unlike these approaches, our method does not rely on human inputs, special camera setup or additional sensors. Similar to our case, Sukhwani et al. [37] computed frame level annotations in broadcast tennis videos, however, they used a dictionary learning method to co-cluster available textual descriptions.

Action Recognition and Segmentation: Deep neural network based approaches such as Two Stream CNN [31], C3D [38], and its derivatives [33, 40] have been instrumental in elevating the benchmark results in action recognition and segmentation. RNNs and LSTMs [14] have also been explored extensively [7, 17] for this task owing to its representation power of long sequence data. Recently, Lea et al. [18] proposed temporal 1D convolution networks variants which are fast to train and perform competitively to other approaches on standard benchmarks for various temporal segmentation tasks.

In the context of sports activity recognition, Ramanathan et al. [25] detected key actors and special events in basketball games by tracking players and classifying events using RNNs with attention mechanism. Ibrahim et al. [15] proposed to recognize multi-person actions in volleyball games by using LSTMs to understand the dynamics of players as well as to aggregate information from various players.

Person Detection and Tracking: An exhaustive survey of this area can be found in [21]. Specific methods for sports videos [19, 30, 41] and especially for handling occlusions [13] have also been proposed in the past. In the context of applications involving player tracking data, Wang et al. [39] used tracking data of basketball matches to perform offensive playcall classification while Cervone et al. [4] performed point-wise predictions and discussed defensive metrics.

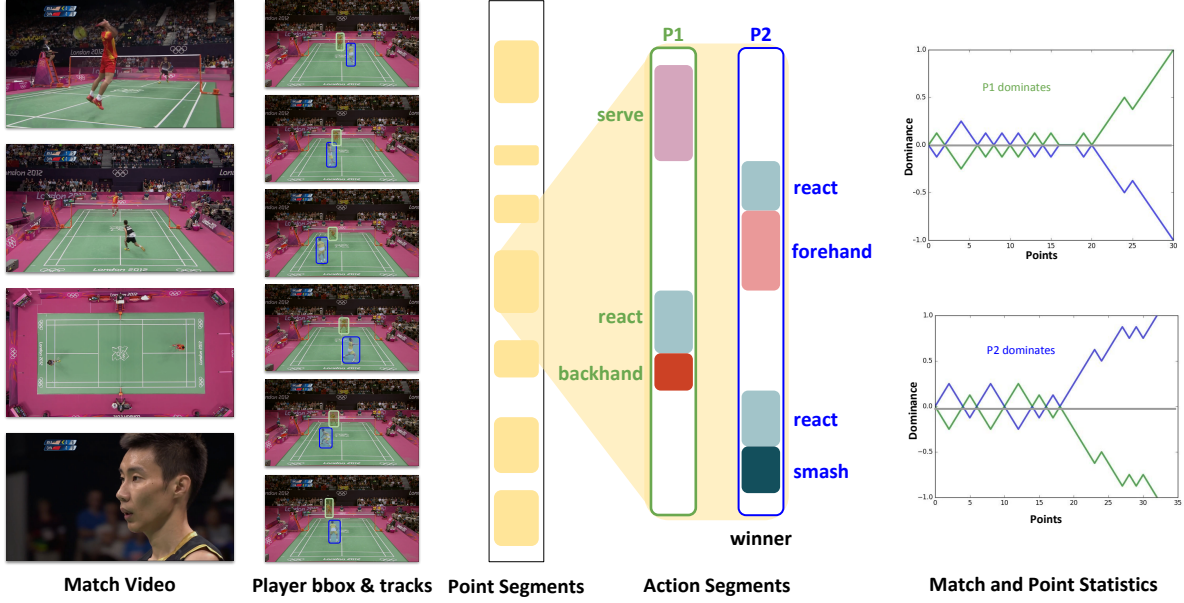


Figure 2: We propose to perform automatic annotation of various gameplay statistics in a badminton game to provide informative analytics. We model this task as players’ detection and identification followed by temporal segmentation of each scored points. To enable deeper analytics, we perform dense temporal segmentation of player’s strokes for each player independently.

Component	Classes	Total	Train	Test
Matches	NA	10	7	3
Players	NA	20	14	6
Player bboxes	2	2988	2094	894
Point segments	2	751	495	256
Strokes	12	15327	9904	5423

Table 1: Various statistics of our Badminton Olympic Dataset. Each match is typically one hour long. Train and test columns represents number of annotations used in respective split for experiments. Note that there is no overlap of players between train and test splits.

3. Badminton Olympic Dataset

We work on a collection of 27 badminton match videos taken from the official Olympic channel on YouTube¹. We focus on “singles” matches played between two players for two or three sets and are typically around an hour long. Statistics of the proposed dataset used in our experiments are provided in Table 1. Please refer to the supplementary materials for the full list of matches and the corresponding broadcast videos. We plan to release our dataset and annotations publicly post acceptance of the work.

Matches: To train and validate our approach, we manually annotate a subset of 10 matches. For this, we select

only one match per player which means no player plays more than one match against any other player. We choose this criteria to incorporate maximum gameplay variations in our dataset as well as to avoid overfitting to any specific player for any of the tasks. We divide the 10 matches into training set of 7 matches and a test set of 3 matches. Note that this setup is identical to leave-N-subjects-out criteria which is followed in various temporal segmentation tasks [8, 24, 33, 35]. Evaluation across pairs of unseen players also emphasize the generality of our approach.

Points: In order to localize the temporal locations of when points are scored in a match, we annotate 751 points and obtain sections that are corresponding to point and non-point segments. We annotate the current score, and the identity of the bottom player (to indicate the court switch after sets/between final set). Apart from this, we also annotate the serving and the winner of all the points in each set for validating outcome prediction.

Player bounding boxes: We focus on “singles” badminton matches of two players. The players switch court after each set and midway between the final set. In a common broadcast viewpoint one player plays in the court near to the camera while the other player in the distant court (see Fig. 1), which we refer to as bottom and top player respectively. We randomly sample and annotate 150 frames with bounding boxes for both players in each match (total around

¹<https://www.youtube.com/user/olympic/>



Figure 3: Representative “strokes” of bottom and top players for each class taken from our Badminton Olympic Dataset. The images have been automatically cropped using bounding boxes obtained from the player detection model. Top player appear smaller and have more complex background than the bottom player, therefore, are more difficult to detect and recognize strokes.

3000 boxes) and use this for the player detection task. The players are occasionally mired by occlusion and the large playing area induces sudden fast player movements. As the game is very fast-paced, large pose and scale variations exist along with severe motion blur for both players.

Strokes: The badminton strokes can be broadly categorized as “serve”, “forehand”, “backhand”, “lob”, and “smash” (refer to Figure 3 for representative images). Apart from this we identify one more class, “react” for the purpose of player’s gameplay analysis. A player can only perform one of five standard strokes when the shuttle is in his/her court while the opponent player waits and prepare for response stroke. After each stroke the time gap for response from other player is labeled as “react”. Also, we differentiate between the stroke classes of the top player and the bottom player to identify two classes per stroke (say, “smash-top” and “smash-bottom”). We also add a “none” class for segments when there is no specific action occurring. We manually annotate all strokes of 10 matches for both players as one of the mentioned 12 ($5 \cdot 2 + 2$) classes.

The “react” class is an important and unique aspect of our dataset. When a player plays aggressively, that allows very short duration for the opponent to decide and react. It is considered to be advantageous for the player as the opponent often fails to react in time or make a mistake in this short critical time. To the best of our knowledge, ours is the only temporal segmentation dataset with such property due to the rules of the game. This aspect is evident in racket sports as a player plays only a single stroke (in a well sepa-

rated playing space) at a time.

Sports videos have been an excellent benchmarks for action recognition techniques and many datasets have been proposed in the past [15, 16, 25, 34]. However, these datasets are either trimmed [16, 34] or focused on team based sports [15, 25] (multi-person actions with high occlusions). On the contrary, for racket sports (multi-person actions with relatively less occlusion) there is no publicly available dataset. Our dataset is also significant for evaluation of temporal segmentation techniques since precise boundary of each action and processing each frame is of equal importance unlike existing temporal segmentation datasets [8, 35] which often have long non-informative background class. Sports videos also exhibit complex sequence of actions (depending on player’s strategy, deceptions, injury etc.) unlike trivial sequences in other domains (e.g. cooking activity follows a fixed recipe) are excellent benchmarks for forecasting tasks.

4. Extracting Player Data

We start by finding the video segments that correspond to the play in badminton, discarding replays and other non-relevant sections. We then proceed to detect, track and identify players across these play segments. Lastly, we recognize the strokes played by the players in each play segment. We use these predictions to generate a set of statistics for effective analysis of game play.

4.1. Point Segmentation

We segment out badminton “points” from the match by observing that usually the camera is behind the baseline during the play and involves no or minimal camera panning. The replays are usually recorded from a closer angle and focus more on the drama of the stroke rather than the game in itself (however, very few points are also recorded from this view), and thus adds little or no extra information for further analysis. We extract HOG features from every 10th frame of the video and learn a χ^2 kernel SVM to label the frames either as a “point frame” or a “non-point frame”. We use this learned classifier to label each frame of the dataset as a “point frame” or otherwise and smoothen this sequence using a Kalman filter.

Evaluation The F1 score for the two classes for optimal parameters (C and $order$) was found to be 95.44%. The precision and recall for the point class were 97.83% and 91.02% respectively.

4.2. Player Tracking and Identification

We finetune a FasterRCNN [27] network for two classes, “PlayerTop” and “PlayerBottom” with manually annotated players bounding boxes. The “top player” corresponds to the player on the far side of the court and while the “bottom player” corresponds to the player on the near side of the court w.r.t to the viewpoint of the camera, and we’ll be using this notation for brevity. For the rest of the frames, we obtain bounding boxes for both the players using the trained model. This approach absolves us from explicitly tracking the players with more complex multi-object trackers. We run the detector for every other frame in a point segment to get the player tracks.

We further find the players’ correspondences across points, as the players change court sides after each set (and also in the middle of the third set). For performing player level analysis it is important to know which player’s statistics we are actually computing. The players wear the same colored jersey across a match and it is dissimilar from the opponent’s jersey. We segment the background from the foreground regions using moving average background subtraction method [12]. We then extract color histogram features from the detected bounding box after applying the foreground mask, and take it as our feature. Now, for each point, we randomly average 10 player features corresponding to the point segments to create 2 player features per point. We cluster the features using a Gaussian Mixture Model into 2 clusters for each match. We then label one cluster as the first player and the other cluster as the second player. This method, although simple, is not extensible to tournaments where both the players are wearing a standard tournament kit, such as Wimbledon in Tennis.

ED-TCN	Metric	d=5	d=10	d=15	d=20
HOG	Acc	71.02	72.56	71.98	71.61
	Edit	76.10	80.52	80.12	79.66
SpatialCNN	Acc	69.19	68.92	71.31	71.49
	Edit	77.63	80.48	80.45	80.40
Dilated TCN	Metric	s=1	s=2	s=4	s=8
HOG	Acc	70.24	68.08	68.25	67.31
	Edit	70.11	70.72	73.68	73.29
SpatialCNN	Acc	69.59	69.75	69.37	67.03
	Edit	59.98	69.46	74.17	71.86

Table 2: We evaluate the player stroke segmentation by experimenting with filter size ‘d’ and sample rate ‘s’ respectively. *Acc* corresponds to per time step accuracy while *Edit* corresponds to edit score.

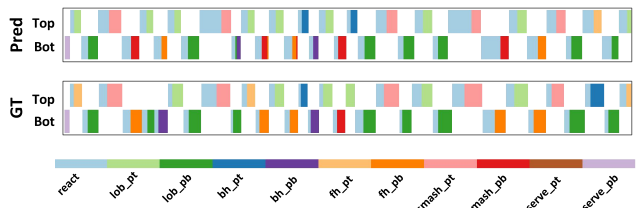


Figure 4: **Stroke Visualization.** For a representative point, segment level strokes obtained from experiments are shown. Each action label has been color coded. (*Best viewed in color*)

Evaluation For evaluating the efficacy of the learnt player detection model, we computed the $mAP@0.5$ values on the test set which were found to be 97.85% for the bottom player, while it was found to be 96.90% for the top player.

For evaluating the player correspondences, we compare the identity assignments obtained from the clusters with the manual annotations of player identity for each point. As our method is unsupervised, we evaluated the assignments for all the points in 10 matches. The method described above yielded us an averaged accuracy of 97.98%, which can be considered adequate for the further tasks.

4.3. Player Stroke Segmentation

We employ and adapt the Temporal Convolutional Network (TCN) variants described by Lea et al. [18] for this task. The first kind, Encoder Decoder TCN (ED-TCN), is similar to the SegNet [2] architecture (used for semantic segmentation tasks), the encoder layers consist of, in order, temporal constitutional filters, non linear activation function and temporal max-pooling. The decoder is analogous to the encoder instead it employs up-sampling rather than pooling, and the order of operations is reversed. The filter count of each encoder-decoder layer is maintained to achieve symmetry w.r.t. architecture. The prediction is then the proba-

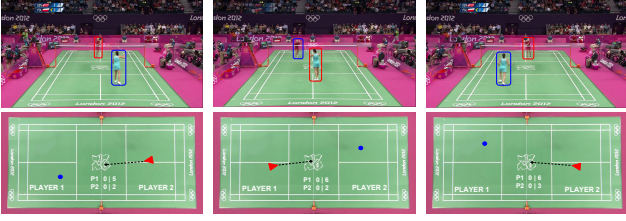


Figure 5: **Detecting Point Outcome.** The serving player is indicated in red. It can be observed that the serving player is usually closer to the mid-line than the receiver who centers himself in the opposite court. Also, the positions of the players in the second point w.r.t. to the first point indicate that the bottom player has won the last point (as the serve is switched). (*Best viewed in color*)

bility of each class per time step obtained by applying the softmax function.

The second kind, Dilated TCN is analogous to the WaveNet [22] architecture (used in speech synthesis tasks). A series of blocks are defined (say B), each containing L convolutional layers, with the same number of filters F_w . Each layer has a set of dilated convolutions with rate parameter s , activation and residual connection that combines the input and the convolution signal, with the activation in the l^{th} layer and the j^{th} block is denoted as $S^{(j,l)}$. Assuming that the filters are parameterized by $W^{(1)}, W^{(2)}, b$ along with residual weight and bias parameters V and e ,

$$\begin{aligned}\hat{S}_t^{(j,l)} &= f(W^{(1)}S_{t-s}^{j,l-1} + W^{(2)}S_t^{j,l-1} + b) \\ S_t^{(j,l)} &= S_t^{(j,l-1)} + V\hat{S}_t^{(j,l)} + e\end{aligned}$$

The output of each block is summed using a set of skipped connections by adding up the activations and applying the $ReLU$ activation, say $Z_t^0 = ReLU(\sum_{j=1}^B S_t^{(j,L)})$. A latent state is defined as $Z_t^{(1)} = ReLU(V_r Z_t^{(0)} + e_r)$ where V_r and e_r are learned weight and bias parameters. The predictions are then given by applying the softmax function on $Z_t^{(1)}$.

To learn the parameters, the loss employed is categorical cross-entropy with SGD updates. We use balanced class weighting for both the models in the cross entropy loss to reduce the effect of class imbalance.

We experiment with two different feature types, HOG and a CNN. Inspired by the use of HOG features by [6] for performing stroke recognition, we extract HOG features. As [18] benchmark using trained Spatial CNN features and the current benchmarks all use convolutional neural networks, we employ the Spatial CNN from the two stream convolution network model [31] to extract features. However, instead of extracting features globally, we instead utilize the earlier obtained player tracks and extract the image region of input scale (454×340) centered at player track centroid

for each time step which are then resized to 224×224 . We then independently extract features for both the players and concatenate the obtained features per frame. The Spatial CNN used is trained on the UCF101 dataset, and we experiment with the output of FC7 layer as our features. The HOG features are extracted with a cell size of 64.

We use the default parameters for training the TCN variants as reported in [18]. We experiment the effect of dilation by setting the sample rate (s) at 1, 2, 4 and 8 fps for Dilated TCN. For ED-TCN we vary the convolutional filter size (say d) of 5, 10, 15 and 20 (setting $s = 2$). We employ acausal convolution for ED-TCN by convolving from $X_{t-\frac{d}{2}}$ to $X_{t+\frac{d}{2}}$. For the Dilated TCN case, we add the term $W^{(2)}S_{t+s}^{j,l-1}$ to the update equation mentioned earlier [18]. Here, X is the set of features per point and t is the time step.

Evaluation We employ the per frame accuracy and edit score metric used commonly for segmental tasks [18], and the results can be seen in Table. 2. We also experimented with causal models by convolving from X_{t-d} to X_t but observed that the performance of those models is not comparable to acausal models and thus did not report those results. The low performance of causal models can be attributed to the fact that badminton is fast-paced and unpredictable in nature. ED-TCN outperforms Dilated TCN which is consistent with benchmarking on other datasets [18]. We can observe that the filter size of 10 is most appropriate for the ED-TCN while the sample rate of 4 is most appropriate for Dilated TCN. From Fig 4, it can be seen that the backhand and forehand strokes are prone to confusion, also smash and forehand strokes. Please refer to the supplementary materials for more exhaustive and detailed results.

5. Detecting Point Outcome

The badminton scoring system is simple to follow and incorporate into our system. At the beginning of the game (score 0 — 0) or when the serving player’s score is even, the serving player serves from the right service court, otherwise, from the left service court. If the serving player wins a rally, they score a point and continues to serve from the alternate service court. If the receiving player wins a rally, the receiving player scores a point and becomes the next serving player.

We exploit this game rule and its relationship to players’ spatial positions on the court for automatically predicting point outcomes. Consider point video segments obtained from point segmentation (Section 4.1), we record predicted players’ positions, and strokes played in each frame. At the start of next rally segment, the player performing the “serve” stroke is inferred as the winner of previous rally and point is awarded accordingly. We, therefore, know the point assignment history of each point by following this proce-

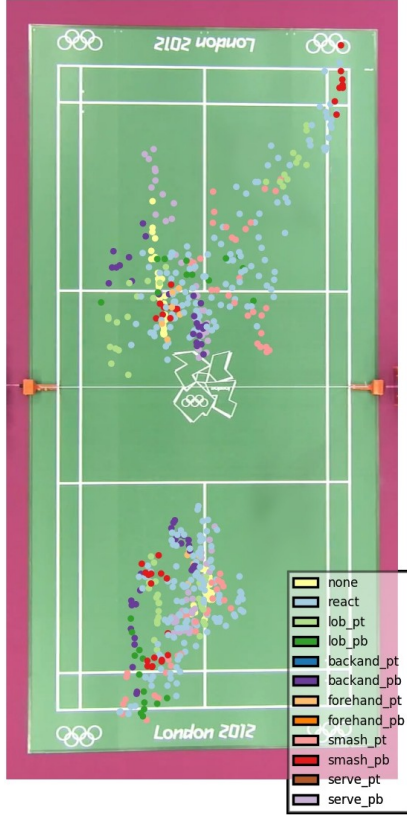


Figure 6: **Point Summary.** We show the frame level players’ positions and footwork around the court corresponding to game play of a single point won by the bottom player. The color index correspond to the stroke being played. Note that, footwork of bottom player is more dense compared to that of top player indicating the dominance of bottom player. (*Best viewed in color*)

ture from the start and until the end of the set. Also, it should be noted that for a badminton game the spatial position of both the serving and the receiving players is intrinsically linked to their positions on the court (See Fig. 5 for a detailed explanation). While similar observations have been used earlier by [42] to detect serve scenes, we detect both the serving player and the winning player (by exploiting the game rules) without any specialized setup.

We formulate the outcome detection problem as a binary classification task by classifying who is the serving player, i.e. either the top player is serving or the bottom player is serving. Thus, we employ a kernel SVM as our classifier, experimenting with polynomial kernel. The input features are simply the concatenated player tracks extracted earlier i.e. for the first k frames in a point segment, we extract the player tracks for both the players to construct a vector of length $8k$.

We varied the number of frames and the degree of polynomial kernel for our experiments and tested on the 3 test

matches as described earlier. We observed that the averaged accuracy was found to be 94.14% when the player bounding boxes of the first 50 frames are taken and the degree of the kernel is six.

6. Analyzing Points

The player tracks and stroke segments can be utilized in various ways for data analysis. For instance, the simplest method would be the creation of a pictorial **point summary**. For a given point (see Fig. 6), we plot the “center-bottom” bounding box positions of the players in the top court coordinates by computing the homography. We then color code the position markers depending on the “action”/“reaction” they were performing then. From Fig. 6, it is evident that the bottom player definitely had an upper hand in this point as the top player’s positions are scattered around the court. These kind of visualizations are useful to quickly review a match and gain insight into player tactics.

We attempt to extract some meaningful statistics from our data. The temporal structure of Badminton as a sport is characterized by short intermittent actions and high intensity [23]. The pace of badminton is swift and the court situation is always continuously evolving, and difficulty of the game is bolstered by the complexity and precision of player movements. The decisive factor for the games is found to be speed [23], and it’s constituents,

- Speed of an individual movement
- Frequency of movements
- Reaction Time

In light of such analysis of the badminton game, we define and automatically compute relevant measures that can be extracted to quantitatively and qualitatively analyze player performance in a point and characterize match segments. We use the statistics presented in Fig. 7 for a match as an example.

1. **Set Dominance** We utilize the detected outcomes and the player identification details to define dominance of a player. We start the set with no player dominating over the other and we define a player as dominating if they have won consecutive points in a set and add one mark to the dominator and subtract one mark from the opponent likewise. We then plot the time sequence to find both “close” and “dominating” sections of a match.

For instance, in Fig. 7, which are statistics computed for a match, it’s apparent that the initial half of the first set was not dominated by either players and afterwards one of the players took the lead. The same player continued to dominate in the second set and win the game.

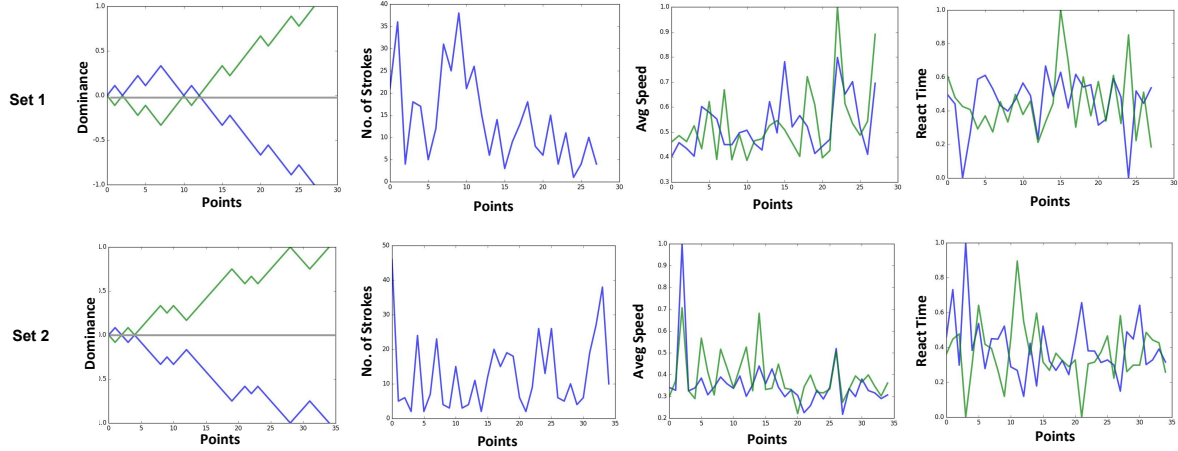


Figure 7: The computed statistics for a match, where each row corresponds to a set. It should be noted that green corresponds to the first player, while blue corresponds to second player. The first player won the match. (*Best viewed in color*)

2. **Number of strokes in a point** A good proxy for aggressive play is the number of strokes being played by the two players. Aggressive and interesting play usually results in long rallies and multiple back-and-forth plays before culminating in a point scored for one or the other players. To approximate, we count the number of strokes in a point. Interestingly, it can be observed in Fig. 7 that the stroke count is higher during the points none of the players are dominating in the match we have taken as example.
3. **Average speed in a point** To find the average speed of the players in a point, we utilize our player tracks. However, displacement of both the players would manifest differently in the camera coordinates. Thus we detect the court lines in the video frame and find the homography with the camera view (i.e. behind the baseline view) of the court. We then use the bottom of the player bounding boxes as proxy for feet and track that point in the camera view. Using a Kalman Filter, we compute displacement and thus speed in the overhead view (taking velocity into account through the observations matrix) and normalize the values. This would act as a proxy for intensity within a point.
4. **Average Reaction Time** We approximate reaction time by averaging the time for react class separately for both the players and then normalizing the values. We assume that the reaction time for the next stroke corresponds to the player who is performing it (See Fig. 4) to disambiguate between the reactions. This measure could be seen as the leeway the opponent provides the player.

7. Discussions

In this work, we present an end-to-end framework for automatic analysis of broadcast badminton videos. We build our pipeline on off-the-shelf object detection [27], action recognition and segmentation [18] modules. Analytics for different sports rely on these modules making our pipeline generic for various sports, especially racket sports (tennis, badminton, table tennis, etc.). Although these modules are trained, fine-tuned, and used independently, we could compute various useful as well as easily understandable metrics, from each of these modules, for higher-level analytics. The metrics could be computed or used differently for different sports but the underlying modules rarely change. This is because broadcast videos of different sports share the similar challenges. We present a comprehensive analysis of errors and failures of our approach in the supplementary materials. We also show a detailed match level stroke, players' positions and footwork visualizations as well as match statistics obtained using our framework along with videos to showcase the information augmentation during game play.

Rare short term strategies (e.g., deception) and long term strategies (e.g., footwork around the court) can be inferred with varying degree of confidence but not automatically detected in our current approach. To detect deception strategy, which could fool humans (players as well as annotator), a robust fine-grained action recognition technique would be needed. Whereas, predicting footwork requires long term memory of game states. These aspects of analysis are out of scope of this work. Another challenging task is forecasting player's reaction or position. It's specially challenging for sports videos due to the fast paced nature of game play, complex strategies as well as unique playing styles.

References

- [1] Dartfish: Sports performance analysis. <http://dartfish.com/>. 1
- [2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *PAMI*, 2017. 5
- [3] G. Bertasius, H. S. Park, S. X. Yu, and J. Shi. Am i a baller? basketball performance assessment from first-person videos. In *Proc. ICCV*, 2017. 2
- [4] D. Cervone, A. DAmour, L. Bornn, and K. Goldsberry. Pointwise: Predicting points and valuing decisions in real time with nba optical tracking data. In *MITSSAC*, 2014. 2
- [5] S. Chen, Z. Feng, Q. Lu, B. Mahasseni, T. Fiez, A. Fern, and S. Todorovic. Play type recognition in real-world football video. In *Proc. WACV*, 2014. 2
- [6] W.-T. Chu and S. Situmeang. Badminton Video Analysis based on Spatiotemporal and Stroke Features. In *Proc. ICMR*, 2017. 1, 2, 6
- [7] A. Fathi and J. M. Rehg. Modeling actions through state changes. In *Proc. CVPR*, 2013. 2
- [8] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In *Proc. CVPR*, 2011. 3, 4
- [9] B. Ghanem, M. Kreidieh, M. Farra, and T. Zhang. Context-aware learning for automatic sports highlight recognition. In *Proc. ICPR*, 2012. 1, 2
- [10] A. Hanjalic. Generic approach to highlights extraction from a sport video. In *Proc. ICIP*, 2003. 2
- [11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 1
- [12] J. Heikkilä and O. Silvén. A real-time system for monitoring of cyclists and pedestrians. *IVC*, 2004. 5
- [13] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *Proc. ECCV*, 2016. 2
- [14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 2
- [15] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori. A hierarchical deep temporal model for group activity recognition. In *Proc. CVPR*, 2016. 2, 4
- [16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. CVPR*, 2014. 4
- [17] C. Lea, A. Reiter, R. Vidal, and G. D. Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *Proc. ECCVW*, 2016. 2
- [18] C. Lea, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks: A unified approach to action segmentation. In *Proc. ECCVW*, 2016. 1, 2, 5, 6, 8
- [19] M. Mentzelopoulos, A. Psarrou, A. Angelopoulou, and J. García-Rodríguez. Active foreground region extraction and tracking for sports video annotation. *Neural processing letters*, 2013. 2
- [20] M. Mlakar and M. Luštrek. Analyzing tennis game through sensor data with machine learning and multi-objective optimization. In *Proc. UbiComp*, 2017. 1, 2
- [21] D. T. Nguyen, W. Li, and P. O. Ogunbona. Human detection from images and videos: A survey. *PR*, 2016. 2
- [22] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 6
- [23] M. Phomsoupha and G. Laffaye. The science of badminton: Game characteristics, anthropometry, physiology, visual fitness and biomechanics. In *Sports Medicine*, 2015. 7
- [24] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Proc. CVPR*, 2012. 3
- [25] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei. Detecting events and key actors in multi-person videos. In *Proc. CVPR*, 2016. 2, 4
- [26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, 2016. 1
- [27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. NIPS*, 2015. 1, 5, 8
- [28] V. Ren, N. Mosca, M. Nitti, T. D’Orazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella. A technology platform for automatic high-level tennis game analysis. *CVIU*, 2017. 1, 2
- [29] L. Sha, P. Lucey, S. Sridharan, S. Morgan, and D. Pease. Understanding and analyzing a large collection of archived swimming videos. In *Proc. WACV*, 2014. 2
- [30] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *Proc. ICCV*, 2011. 2
- [31] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proc. NIPS*, 2014. 2, 6
- [32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2014. 1
- [33] S. Singh, C. Arora, and C. Jawahar. First person action recognition using deep learned descriptors. In *Proc. CVPR*, 2016. 1, 2, 3
- [34] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: a dataset of 101 human actions classes from videos in the wild. 2012. 4
- [35] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proc. UbiComp*, 2013. 3, 4
- [36] M. Sukhwani and C. Jawahar. Tennisvid2text: Fine-grained descriptions for domain specific videos. In *Proc. BMVC*, 2015. 2
- [37] M. Sukhwani and C. Jawahar. Frame level annotations for tennis videos. In *Proc. ICPR*, 2016. 2
- [38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. ICCV*, 2015. 2
- [39] K.-C. Wang and R. Zemel. Classifying nba offensive plays using neural networks. In *MITSSAC*, 2016. 2
- [40] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proc. CVPR*, 2015. 1, 2
- [41] F. Yan, J. Kittler, D. Windridge, W. Christmas, K. Mikolajczyk, S. Cox, and Q. Huang. Automatic annotation of tennis games: An integration of audio, vision, and learning. *Image and Vision Computing*, 2014. 1, 2
- [42] F. Yoshikawa, T. Kobayashi, K. Watanabe, and N. Otsu. Automated service scene detection for badminton game analysis using CHLAC and MRA. 2, 7