BADMINTON STROKE CLASSIFICATION USING LONG SHORT TERM MEMORY

Avinav Jain   Shubham Agrawal   Gaurav Singh Chauhan   Sai Shruti I   Preety Singh

The LNM Institute of Information Technology, Jaipur, Rajasthan, India
{avinavjain.y19, shubhamagrawal.y19, gauravchauhan.y19, saishrutii.y19, preety} @lnmiit.ac.in
https://www.lnmiit.ac.in/

ABSTRACT

In this paper, we have classified three different types of strokes played in the game of badminton. For the purpose of this study, we have created our own dataset. We have built a time-based sequential model to recognize these shots by tracking the movements of various key points of the player's body. Using this approach, we have achieved an accuracy of 89.49 percent in the classification of three strokes in the game of badminton, namely Clear shot, Serve, and Smash.

Index Terms— Badminton, Sequential Model, LSTM, Keypoints Extraction, Cross-validation

1. INTRODUCTION

SPORTS are an essential part of our lives, and they can be enjoyed by everyone. Automated action recognition can be very useful in this field, for the purpose of analyzing the game and identifying different types of tactics used by players and types of shots played in the game [9]. This can be further extended to identifying a player's weaknesses and strengths which can help him improve his game. A player's posture while playing a shot in racket sports, such as tennis or badminton, speaks a lot about his skill level. So detecting the correctness of the player's posture while playing various shots, can help coaches to detect inadequacies and to plan training accordingly [13,14]. Sports fans are also interested in the statistics of the players.

The use of deep learning in recognition and classification techniques in sports has gained popularity in recent years [10,11,12]. In games like cricket, computer vision and machine learning has become an integral part, as the outcome of someone being *out* can be predicted by tracing the probable trajectory of the ball. Similarly, in tennis, trained machine models are used to tell if the ball played is *fair* or *foul*. In football, it is used to detect the slightest touch between the player and the ball, to detect if it is a call for a *penalty* shootout. Match prediction is also one of the major applications of the use of artificial intelligence in the field of sports. By taking data from the past matches, the system predicts the probability of winning for each team, on the basis of the current match scenario.

Badminton is one of the fastest racket sport and a major sport at global level. However, despite its popularity, there is limited research in this game due to non-availability of a public dataset. In this work, we have created our own dataset of people playing three different kinds of strokes on the badminton court which is available publicly for research purposes. For the purpose of this study, we have taken three classes of shots and used a time-based sequential model to predict the shot played by the player on the basis of his/her body movements which are detected by observing the changes in the position of the body joints in space. The dataset used for this study consists of 328 videos per class.

The dataset is divided into three classes of strokes, namely *Clear*, *Serve*, and *Smash*. The trained model is used in predicting the number of shots of each type played during a match of badminton. The main contributions of our research are as follows:

- Introduced a new dataset comprising of three classes of strokes {Clear, Smash, Serve}
- Built a sequential model to classify and recognize these shots.

The rest of the paper is organized as follows: Section —

2. RELATED WORK

In this section, we present few related works the use of action recognition and classification in the field of sports [1, 5, 8]. In [6], Ghosh et al. proposed a generic pipeline for the analysis of racket games by using action recognition, segmentation, and object detection as sub-modules of their research. In this work, they have provided an analysis of the broadcast videos of badminton matches. Having detected players, points, and strokes for each frame in a match, they tried computing understandable metrics like player's reaction time, dominance, positioning and footwork around the court, etc. Their player detection model works by extracting color histogram features and using the gaussian mixture model, to cluster the image into two groups, each representing a player. This model has a mAP@0.5 value of 97.85% for the bottom player, and 96.90% for the top player. Wherein, the bottom player is the one closer to the camera, whose back is visible, and the top player refers to the player who is far away from the camera, whose front is visible and behind the net. To compute their metrics, they have also built a model for classifying a point frame to that of a non-point frame, extracting the HOG features from every 10th frame of the video and building a χ^2 kernel SVM to label a frame as point and non-point frame, which works with an average F1-score of 95.44%.

In [2], Chu et al. proposed a framework to classify the strategy of an individual player in the game of badminton as offensive or defensive. For this purpose, they used various techniques to detect court lines, identify players by subtracting background, and classify players' postures into six categories of stroke types. Finally, taking the sequence of observations of stroke type and player's movements, they predicted the strategy of the gameplay and achieved the average strategy classification accuracy of 70% using a cross-validation model. RNNs and LSTMs have also been explored extensively in this field [15, 16, 17]

In [3], Fok et al. have presented a framework to recognize various styles of swimming, such as butterfly, freestyle, etc. by feeding the posture key points to a neural network model which has Dropout and LSTM layers to facilitate over-fitting and vanishing gradient issues. Using their approach, they were able to get 92.9% test accuracy. In [7], Mora et al. classified strokes in the game of tennis into four action groups and trained a 3-layered LSTM model on an independent dataset. They reported an accuracy of 84.10% when tested on a mixed set of matches of amateurs and professionals.

In [18], Kumar et al. worked on cricket. They used the optical flow technique to identify whether the ball was hit by a batsman and also identified if the shot played is of a defensive type. Because of the small dataset, they have used the K-fold cross-validation spitting technique and obtained an accuracy of 80%. In [4], Tsunoda et al. used multi-person centered features and temporal dynamics of features, to recognize various futsal plays in the game of soccer recorded from multiple cameras in multi-view. They have integrated one more state in the general LSTM, called as *Keeping state*, which is externally controlled. Their proposed method with RGB image and meta-information inputs yielded an accuracy of 70.90%.

3. PROPOSED METHODOLOGY

A block diagram of our proposed methodology is given in Figure 1 and explained in the subsections.

3.1. Dataset Collection

We have created our own dataset by recording shots and published collected and augmented dataset at this link: https://github.com/sam-2200/Badminton-Dataset/. Our videos are labeled into three classes, namely, *Clear*, *Serve*, and *Smash*. For the purpose of recording, we have used Microsoft Xbox Kinect One Sensor and recorded the videos at 30fps. To simulate behind the baseline view, we placed the sensor around 80cm above the ground, behind the player, and parallel to the net. The player whose back is visible is more in view as compared to the other player across the net. To record the front view, we placed the sensor directly under the net at a similar height. —We have collected 328 videos for each class of shot.– In our dataset, there are five files per data point - a color video, a depth video, a raw depth video, a skeleton points video, and a JSON file with the information about the skeleton joints coordinates. These files are more clearly explained below.

Shot	Number of Videos	Average frames per video
Clear	328	9.31097
Serve	328	10.90548
Smash	328	10.42073

Table 1. Dataset Details

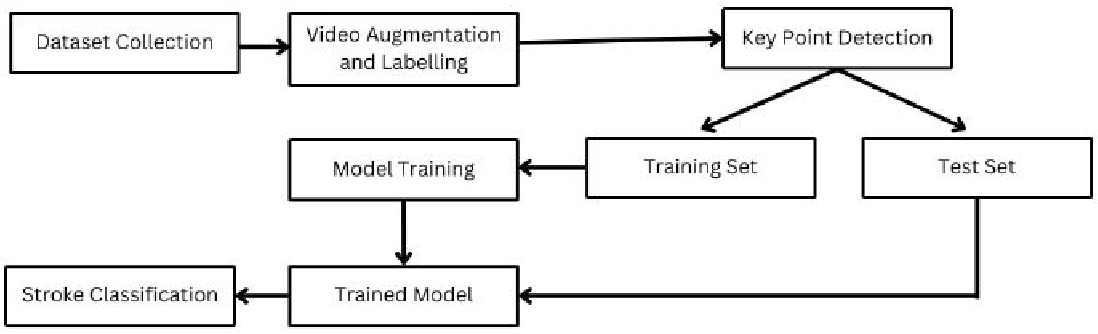

Fig. 1. Block diagram of proposed methodology.

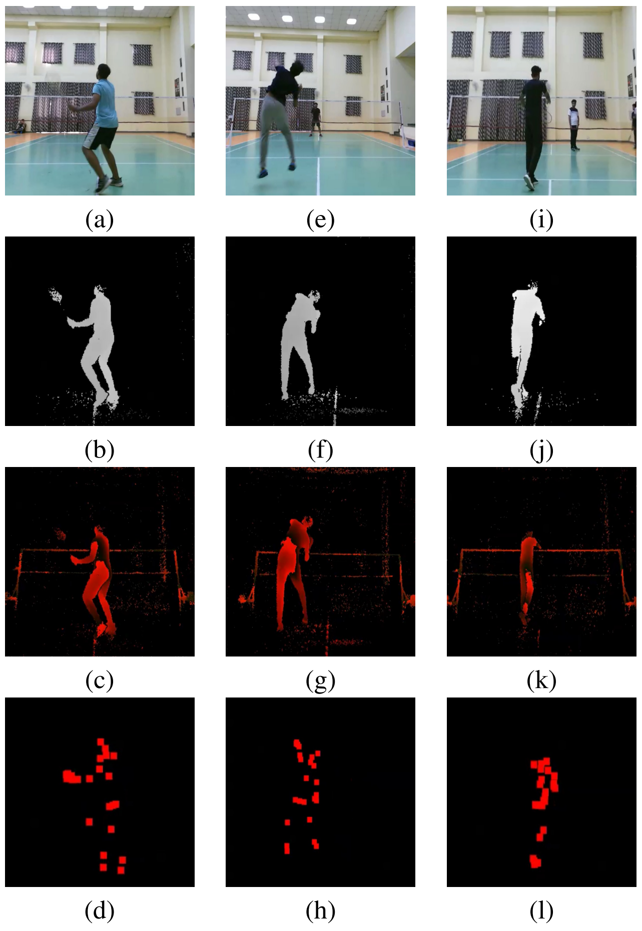

Figure 2: *Images(a-d) Clear shot RGB View, Depth View, Raw Depth View and Skeleton View respectively*

Images(e-h) Serve shot RGB View, Depth View, Raw Depth View and Skeleton View respectively

Images(i-l) Smash shot RGB View, Depth View, Raw Depth View, and Skeleton View respectively

The Kinect contains two cameras with different resolutions: a color (RGB) camera and a depth camera. The color camera has a resolution of 1920×1080. The depth camera has a resolution of 512×424. The color coordinates in 2D and depth ones accommodate this. The Kinect's camera coordinates in 3D utilize the IR beam to pinpoint the three-dimensional positions of the joints in the real world. Kinect provides joint orientation using quaternions, which are a four-dimensional representation of the 3D orientation.

3.1.1. *Understanding Depth*

The Kinect sensor provides the depth stream data as an extension of the depth field of view and offers raw depth data in 16-bit gray-scale format. It applies several algorithms to the data to provide more information that shows how far away each pixel in a frame is. The depth pixel records, in millimeters, the distance of objects in front of it. In the depth sensor view, 2D coordinates represent the data.

3.1.2. *Understanding Skeleton*

The Kinect skeleton returns human body joints. The joints are numbered from 0 to 24. An example is shown in Figure 2. The 11 attributes of each joint are as follows:

- *Color* (x, y) are the joint coordinates on the color camera picture. A color space point is a two-dimensional point on a color picture. So a color space position is a row/column location of a pixel on the picture, where x=0, y=0 corresponds to the top left pixel, and x=1919, y=1079 refers to the rightmost in the bottom row. For color coordinates, the sensor delivers a value belonging to [0,1] which is obtained by dividing the value by the resolution of the sensor.

- *depth* (x, y) is that of the joint on the depth camera picture. Depth space is the name given to a 2D place on a depth picture. The pixel's location on the image is the same as that of a coordinate in a matrix, where (0,0) denotes the leftmost pixel in the top row, and (511,423) corresponds to the rightmost pixel in the bottom row. Same as the color coordinates, the sensor outputs a value belonging to [0,1] for the depth camera as well.

- *camera* (x, y, z) utilizes the IR beam to pinpoint the three-dimensional positions of the joints in the real world. The same is utilized in projects of three-dimensional joint placements. These are treated in a separate way from the two-dimensional color and depth coordinates. The 3D coordinate system employed by the sensor is defined in the following way:
  - The center point of the infrared sensor is the coordinate (0,0,0)
  - With reference to the sensor's point of view, the x coordinate increases towards left
  - With reference to the sensor's tilt, the y coordinate increases upwards
  - The direction of the face of the sensor in the positive direction for the z coordinate

  The range of the sensor to track joint points is around 1/2 meters to 4.5 meters, however, if we decrease the distance between the body and sensor to less than 1 meter, it skeleton view is not much particular. whereas, the depth sensor has a range that extends to 8 meters. As a result, the cameraZ value typically belongs to [1.5, 4.5].

  Both the x and y coordinates have their domain as the real number set. Depending on the position of the joint, they can take both positive and negative values. As stated earlier, the x coordinate increases in the left direction, hence any point, which is to the right of the sensor takes a negative x value. Similarly, the y coordinate increases upwards, with respect to the tilt of the sensor, hence below the sensor, the y coordinate takes a negative value. Y range depends also on the distance of the object from the sensor, however, it can detect heights of up to five meters. The camera space's three values(x, y, z) are scaled on similar distance units to maintain proportions, and the unit used is the meter.

- *orientation* (x, y, z, w) Kinect provides joint orientation using quaternions, which are a four-dimensional representation of the 3D orientation, and they must be translated to be helpful. [Quaternions] is a type of 3D space orientation that is used to prevent gimbal-lock issues caused by employing Euler angles for rotation. They need to be converted into a matrix form.

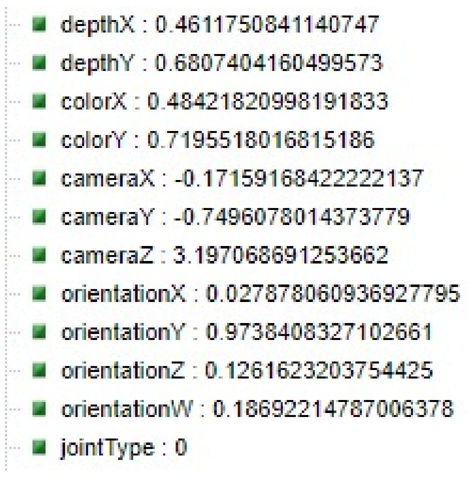
```
■ depthX : 0.4611750841140747
■ depthY : 0.6807404160499573
■ colorX : 0.48421820998191833
■ colorY : 0.7195518016815186
■ cameraX : -0.1715916842222137
■ cameraY : -0.7496078014373779
■ cameraZ : 3.197068691253662
■ orientationX : 0.027878060936927795
■ orientationY : 0.9738408327102661
■ orientationZ : 0.1261623203754425
■ orientationW : 0.18692214787006378
■ jointType : 0
```

Fig. 2. Example of the coordinate space of a joint point.

While the Kinect sensor provides both RGB and depth videos, we have used only the RGB videos so that the experimental results can be extended without depending on any special camera sensor. Also, as in a professional match, usually, the whole game is shown from behind the baseline, we have considered the back recordings of the player for our experiments.

3.2. Video Augmentation and Labelling

A deep learning model requires a large dataset for training. To accommodate this, we have applied augmentation on videos, by applying slight contrast changes, flipping frames, and applying slight rotations or blurring frames. This helped in increasing the data many times. The augmented dataset is also available publicly. The total number of videos after augmentation is 5516.

3.3. Keypoint Detection

We have extracted 33 human body key points (refer Figure 3) per frame by applying a keypoint detection model on the augmented dataset of RGB videos. These key points are nothing but three-dimensional coordinates of human joints.

The whole frame may have a lot of details that may not be needed to predict the type of shot played. Moreover, using the whole video instead of extracting key points from each frame will require extra computation. For these reasons, the extraction of key points is done. Corresponding to each feature point we have three dimensions representing its spatial coordinates. This results in a total of 33 x 3 feature points per frame.

3.4. Classification Model

We have built a time-based sequential model to classify these badminton strokes into pre-defined classes. We have used

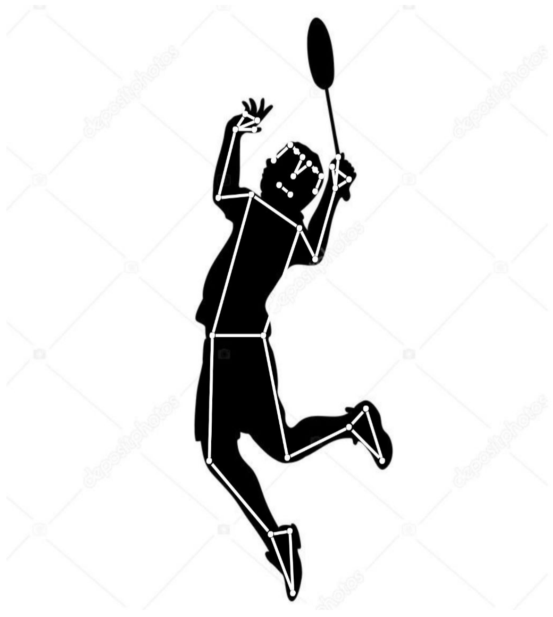

Fig. 3. Extracted keypoints from a human body

the Long Short Term Memory (LSTM) architecture as our deep learning classification model to capture important features that may be present during any part of the stroke play. Our dataset is divided into two parts, 90 percent for the training set, and 10 percent for the test set using a stratified split. During model training, to further increase the training data, we have used a 5-fold cross-validation method on the training data. This serves the purpose of using the whole data in training, as well as in validation in a ratio of $80 : 20$.

3.5. Evaluation Metrics

To evaluate the performance of our model on the test set, we have used the following metrics:

$$Precision\ for\ Label\ A = \frac{TruePositives\ of\ label\ A}{TotalpredictedinLabelA}$$

$$Recall\ for\ Label\ A = \frac{TruePositives\ of\ label\ A}{Total\ test\ items\ with\ label\ A}$$

$$Accuracy\ of\ Model = \frac{\sum_{c\in A} TruePositives(c)}{Total\ Items\ in\ testset}$$

$$where\ A \in \{Clear, Serve, Smash\}$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$SupportoflabelA = TotalItemsWithLabelAInTestSet$$

4. EXPERIMENTS AND RESULTS

For the purpose of collecting the dataset, we used Microsoft Xbox Kinect One Sensor, which provided us with an RGB video, a depth heat map, and skeleton points of the player for each shot played. For our current experiments we have used only the RGB videos. After augmenting our videos, we used MediaPipe Keypoints HolisticsModel to extract the key points in the augmented dataset.

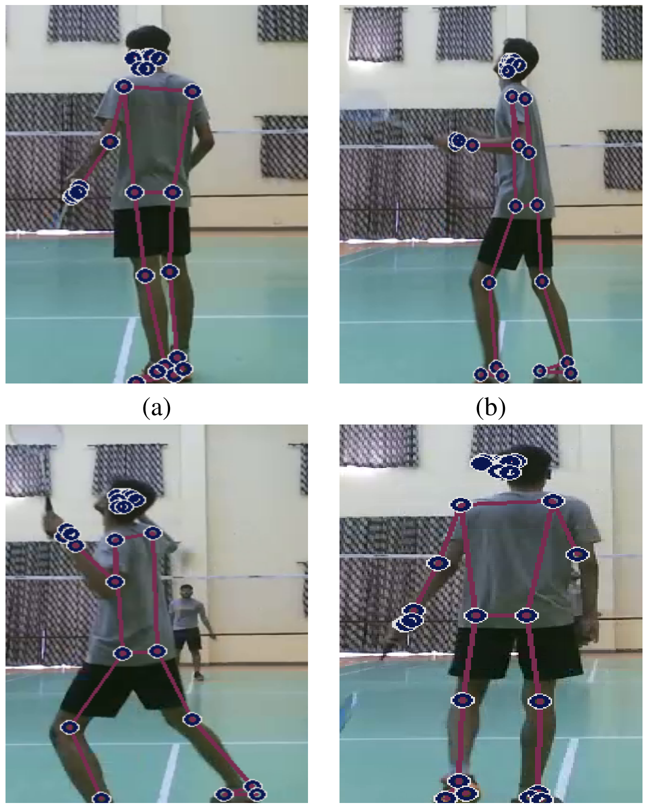

Figure 7: *Images(a-d) Shows plotted keypoints on a video of Clear shot*

As the number of frames varied in different videos, we fixed the number of frames to 15, after carefully observing that in 15 frames almost all the shots are captured. Those videos which exceeded 15 frames generally contain some clip after the shot is played. We applied post-padding to those videos which had frames less than 15.

The keypoint sequences —– of 15 frames each were fed to an LSTM model to train the model to classify the strokes into three classes, *Clear*, *Serve*, and *Smash*. The model uses Adam

Layer Type	Output Shape	Param Count
Time Distributed	(None, 15, 225)	0
LSTM	(None, 15, 512)	1511424
LSTM	(None, 15, 256)	787456
LSTM	(None, 128)	197120
Dense	(None, 64)	8256
Dense	(None, 32)	2080
Dense	(None, 3)	99
Total Parameters	–	2,56,435

Table 2. Model Architecture

	Precision	Recall	F1-score	Support
Serve	0.95	0.94	0.94	194
Smash	0.82	0.95	0.88	186
Clear	0.93	0.78	0.85	172

Table 3. Evaluation metrics for different shots of the game.

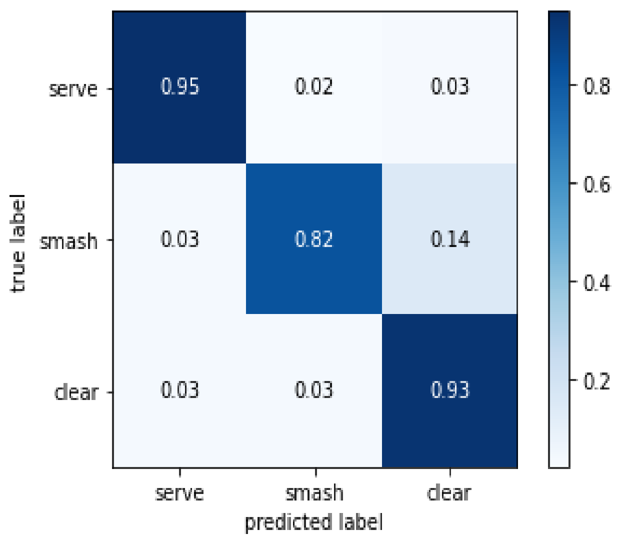

Fig. 4. Confusion matrix.

Optimizer for calculating the step of gradient descent. The architecture of the model is shown in Table 2. The model was trained with 5-fold cross-validation. We obtained an overall accuracy of 89.49% and support of 552. The other evaluation metrics for the different shots are shown in Table 3. The confusion matrix obtained by evaluating the model on —test set—- is shown in Figure 4.

As seen in the confusion matrix, our model works best on serve, seconded by clear shot. Some of the smash shots are getting classified into clear shots, because of the close resemblance of the posture of the player while playing these shots. This can be improved by using more features such as the trajectory of the shuttle cock, together with the player's body movements, to further increase the accuracy of our model.

5. CONCLUSION

In this paper, we have introduced a new labeled dataset, containing different types of shots played in the game of badminton. Further, we have used a time-distributed deep learning model to do stroke classification into three classes, namely Clear, Serve, and Smash. We were able to achieve 89.49 percent accuracy on the test set.

In the future, this study can be further extended by introducing more classes of strokes, and the dataset can also be expanded accordingly. Also, training the model to detect when no shot is played, that is a class called "others" will also help increase the use case of this study.

6. REFERENCES

[1] K.W. Ban, J. See, J. Abdullah, and Y. P. Loh. 2022. BadmintonDB: A Badminton Dataset for Player-specific Match Analysis and Prediction. In Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports (MMSports '22). Association for Computing Machinery, New York, NY, USA, 47–54. https://doi.org/10.1145/3552437.3555696

[2] W. T. Chu and S. Situmeang. 2017. Badminton Video Analysis based on Spatiotemporal and Stroke Features. In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (ICMR '17). Association for Computing Machinery, New York, NY, USA, 448–451. https://doi.org/10.1145/3078971.3079032

[3] Wilton W. T. Fok, Louis C. W. Chan, and Carol Chen. 2018. Artificial Intelligence for Sport Actions and Performance Analysis using Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM). In Proceedings of the 2018 4th International Conference on Robotics and Artificial Intelligence (ICRAI 2018). Association for Computing Machinery, New York, NY, USA, 40–44. https://doi.org/10.1145/3297097.3297115

[4] T. Tsunoda, Y. Komori, M. Matsugu and T. Harada, "Football Action Recognition Using Hierarchical LSTM," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 155-163, doi: 10.1109/CVPRW.2017.25.

[5] Kulkarni, K. M., and Shenoy, S. (2021). "Table Tennis Stroke Recognition Using Two-Dimensional Human Pose Estimation" CVPR Sports Workshop 2021 arXiv. https://doi.org/10.48550/arXiv.2104.09907

[6] A. Ghosh, S. Singh and C. V. Jawahar, "Towards Structured Analysis of Broadcast Badminton Videos,"

2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 296-304, doi: 10.1109/WACV.2018.00039.

[7] S. V. Mora and W. J. Knottenbelt, ”Deep Learning for Domain-Specific Action Recognition in Tennis,” 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 170-178, doi: 10.1109/CVPRW.2017.27.

[8] M. Sharma, M. Lamba, N. Kumar and P. Kumar. (2021). Badminton match outcome prediction model using Naïve Bayes and Feature Weighting technique. Journal of Ambient Intelligence and Humanized Computing. 2020. 10.1007/s12652-020-02578-8.

[9] L. Wang, Y. Qiao, X. Tang. (2015). Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors. 10.1109/CVPR.2015.7299059.

[10] K. He, X. Zhang, S. Ren and J. Sun, ”Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[11] S. Ren, K. He, R. Girshick and J. Sun, ”Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

[12] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In ICLR, 2014 arXiv. https://doi.org/10.48550/arXiv.1409.1556

[13] M. Mlakar and M. Luštrek. 2017. Analyzing tennis game through sensor data with machine learning and multi-objective optimization. In Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers (UbiComp ’17). Association for Computing Machinery, New York, NY, USA, 153–156. https://doi.org/10.1145/3123024.3123163

[14] V. Renò, N. Mosca, M. Nitti, T. DOrazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella. (2017). A technology platform for automatic high-level tennis game analysis. Computer Vision and Image Understanding. 159. 10.1016/j.cviu.2017.01.002.

[15] B. L. Bhatnagar, S. Singh, C. Arora, and C. V. Jawahar. Unsupervised learning of deep feature representation for clustering egocentric actions. In IJCAI, 2017.

[16] A. Fathi and J. M. Rehg, ”Modeling Actions through State Changes,” 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2579-2586, doi: 10.1109/CVPR.2013.333.

[17] Lea, C., Reiter, A., Vidal, R., and Hager, G. D. (2016). Segmental Spatiotemporal CNNs for Fine-grained Action Segmentation. In ECCVW, 2016. arXiv. https://doi.org/10.48550/arXiv.1602.02995

[18] A. Kumar, J. Garg, and A. Mukerjee. (2015). Cricket activity detection. International Image Processing, Applications and Systems Conference, IPAS 2014. 10.1109/IPAS.2014.7043264.