Badminton Stroke Classification using Long Short Term MEmory and Creation of Various Shots played in Badminton

Project report submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Computer Science Engineering

by

Avinav Jain - 19UCS021
Shubham Agarwal - 19UCS073
Sai Shruti I - 19UCC088
Gaurav S Chauhan - 19UCS246

Under Guidance of
Dr. Preety Singh

LNMIIT
The LNM Institute of
Information Technology

Department of Computer Science Engineering
The LNM Institute of Information Technology, Jaipur

August 2022

The LNM Institute of Information Technology
Jaipur, India

CERTIFICATE

This is to certify that the project entitled “Badminton Stroke Classification using Long Short Term Memory and Creation of Various Shots played in Badminton” , submitted by Avinav Jain (19UCS021), Shubham Agarwal (19UCS073), Sai Shruti I (19UCC088) and Gaurav S Chauhan (19UCS246) in partial fulfillment of the requirement of degree in Bachelor of Technology (B. Tech), is a bonafide record of work carried out by them at the Department of Computer Science Engineering, The LNM Institute of Information Technology, Jaipur, (Rajasthan) India, during the academic session 2022-2023 under my supervision and guidance and the same has not been submitted elsewhere for award of any other degree. In my opinion, this report is of standard required for the award of the degree of Bachelor of Technology (B. Tech).

Date	Adviser: Name of BTP Supervisor

Acknowledgments

First and foremost, we would like to express our heartfelt gratitude to our mentor, the distinguished Dr. Preety Singh, who came up with the idea and under whose direction we were able to complete the project. We're also grateful to our college for providing us with a stress-free and positive environment. We would like to show our sincere gratitude towards Mr. Manoj Sain for sharing his knowledge on the use of kinect and for providing us with crucial understanding for uploading our dataset paper. We owe a debt of gratitude to everyone participating in this project because it would not have been possible to complete the project on time without their inspiration and insightful suggestions.

Abstract

Video data for sports events are available for many major tournaments but most of them are not available publicly for research purposes and remain inaccessible. In our work, we have used Microsoft Xbox One Kinect Sensor and Kinectron for recording our dataset for the purpose of action classification of different strokes in the game of badminton.

Besides that, in recent years, a lot of work has been done on human activity classification. Automatic activity recognition can be used to find abnormal activities and alert the authorities regarding some suspicious observations. Also, it can be used in designing video games that can take video inputs. Healthcare facilities can also benefit from this technology, as it can help patients in their rehabilitation.

In this paper, we have picked up three different types of strokes played in the game of badminton, namely Clear shot, Smash, and Serve, and built a time-based sequential model to classify and recognize these shots by tracking the movements of various key points of the player's body. For the purpose of this study, we have created our own dataset, by recording the strokes played by students on a badminton court. Also, we have used the trained model to count the number of each kind of shot played in a game of badminton.

Contents

Chapter 1

Introduction

Sports are an essential part of our lives, and they can be enjoyed by everyone. Whether you're a sports fan or not, there's something for everyone in the world of sports. Automated action recognition can be very useful in this field, for the purpose of identifying different types of tactics a player makes. This can be further extended to identifying a player's weaknesses and strengths which can help him improve his game. Deep Learning's recognition and classification techniques, are getting popular day by day. Sports fans also love players' statistics on the basis of their career for relating closely with the player and games. In games like cricket, computer vision has become an integral part, as the outcome of someone being "out" is predicted by tracing the probable trajectory of the ball, during an event, when it is stopped by the leg of the batsman. Also, in Tennis, computer vision is 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 AI in the field of sports. By taking supervision from past matches, the system predicts on the basis of the current match scenario, the probability of winning each team. 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 how correct is the player's posture while playing various shots, can help coaches to detect where their student is lagging, and to put weight on the particular training[13,14].

1.1 The Area of Work

We have chosen action recognition and classification in the game of badminton as our main area of work.

Badminton is the fastest racket sport and is also a major sport in the Olympics. Despite its popularity, not many people have taken interest in badminton as the protagonist of their study. There is still a lot of scope for study in the field of the use of computer vision in the game of badminton. Most of the work that has been done in the field of stroke classification in this game uses a private dataset, which limits the scope of its extension. Some studies suggest building a model to make predictions on the basis of the trajectory of the shuttlecock, which does not provide much accuracy if the video is available from behind the baseline view, which is the case for most professional matches.

1.2 Problem Addressed

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 class of shots played by the player on the basis of his bodily movements. Body movements are detected by observing the changes in the position of coordinates of body joints in space. The dataset used for this study consists of around 700 videos each covering a shot. The dataset is divided into three classes of strokes, namely Clear, Serve, and Smash. Finally, the model is used in predicting the number of shots per class played during a match of badminton.

Chapter 2

Literature Review

Many studies are published on the use of action recognition and classification in the field of sports.

2.1 Badminton Video Analysis based on Spatiotemporal and Stroke Features

The majority of sports events that are aired give viewers access to game statistics that can be used to plan a gameplay strategy, enhance player performance, or make it easier to find a sport game's key points of interest. For videos of televised badminton, few investigations have, however, been suggested. Several visual analysis approaches are combined in this work to identify the court, identify players, categorise strokes, and categorise the player's strategy. Based on visual analysis, they are learning a little bit about the typical playstyle of a certain player. Based on classification results, study is to assess the effectiveness of stroke and strategy classifications as well as the presentation of game statistics.

2.2 Towards Structured Analysis of Broadcast Badminton Videos

This study provide an end-to-end system for automatic analysis of broadcast badminton videos. Pipeline is built using pre-made object detection, action recognition, and segmentation components. These modules are the foundation for the analytics that make their pipeline generic for a variety of sports, especially racket sports (tennis, badminton, table tennis, etc.). This could combine numerous valuable and simple-to-understand metrics from each of these modules, despite the fact that they are trained, optimised, and used independently, for higher-level analytics. The core modules hardly ever change, even though the stats may be calculated or applied differently for various sports. This is due to the fact that challenges in broadcast footage of many sports are comparable.

Rare short-term tactics (like deceit) and long-term tactics (like footwork around the court) can be inferred with varied degrees of confidence, but they are not automatically detected in our current method. A powerful, fine-grained action identification technique would be required to identify deception tactics that could trick humans (players and annotators). While anticipating footwork calls for long-term game state memory. These analytical facets are not covered by the work at hand. Predicting a player's response or position is another difficult task. Sports videos present a unique set of challenges because of the fast-paced game play, intricate strategy, and distinctive playing styles.

2.3 Artificial Intelligence for Sport Actions and Performance Analysis using Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM)

The Human Action Recognition (HAR) technology is currently gaining popularity. In order to reduce the amount of labour required to provide protection to the public, including public safety and criminal prevention, this project built a HAR system. This study uses a deep learning network with a recurrent neural network and long short-term memory (LSTM) to classify various types of activities and their performance from dynamic video motion of sporting events. For efficiency and memory savings, it might divide up different kinds of human movements into smaller groups using fewer video frames. The accuracy attained to date is up to 92.9%, however there is still room for improvement.

It is recommended that the classifier can be further enhanced by adding more frames for analysis and using more cameras to capture the posture position in order to increase the model's accuracy and decrease the error in the confusion matrix. These actions can significantly expand the datasets available for training and improve the model's accuracy. LSTM and bidirectional flow are two more neural network modules that can be used. The use of recurrent neural networks (RNN) with long short-term memories (LSTM) for athletic actions and performance analysis is seen to be very promising. Numerous possible uses exist, including public security, performance analysis, sport instruction, and sport safety.

Chapter 3

Proposed Work

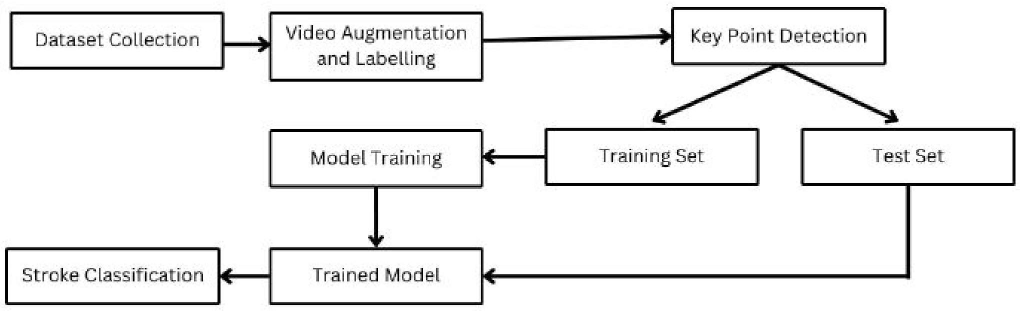


FIGURE 3.1: Block Diagram of Proposed Method

3.0.1 Dataset Collection

We have recorded the shots played by around ten students across multiple days from both front and back. For the purpose of recording, we have used Microsoft Xbox Kinect One Sensor. To stimulate behind the baseline view, we placed the sensor around 80cm above the ground, behind the player and parallel to the net. To record the front view, we placed the sensor directly under the net with a similar height. We have collected – videos for clear shot, – videos for smash, and – videos for serve.

For the purpose of our generalization, in this study, we have used only the RGB videos, so that, the study could be extended in future, without depending on some special camera sensor. Also, in this study, we have considered the back recordings, as, in a professional match, the whole game is shown from behind the baseline, and the player whose back is visible is more in view as compared to the other far away player.

The kinect sensor is placed behind the player and parallel to the net. This allows the sensor to catch footage of the player performing a stroke from behind the baseline view. To assist this, the sensor was positioned on the table about the height of 80 cm, allowing the player's entire body to be caught during games. More than eight players are captured on separate days when capturing the dataset for generalisation purposes. Generalization is essential for constructing scalable models. To distinguish strokes done by different badminton players, our training set needed to be diverse. Players were chosen in such a way that there was a range of heights and years of expertise playing badminton.

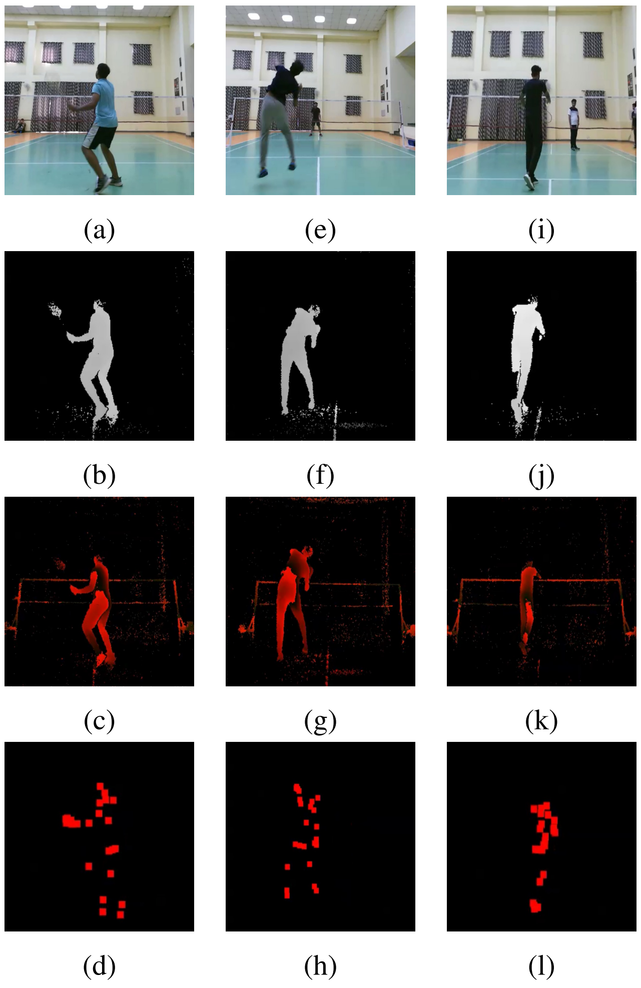


Figure 2: *Images(a-d):Clear shot RGB View, Depth View, Raw Depth View and Skeleton View respectively*
Images(e-h): Serve shot RGB View, Depth View, Raw Depth View and Skeleton View respectively
Images(i-l): Smash shot RGB View, Depth View, Raw Depth View, and Skeleton View respectively

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.

3.0.1.1 Understanding Kinect and Kinectron

The Microsoft Kinect sensor is an XBox and Windows PC attachment that acts similarly to a camera. It does, however, produce a depth map in addition to an RGB picture. The Kinect measures distance from the sensor for each pixel observed by the sensor. This makes a number of computer vision issues easier and more enjoyable, such as background removal and blob recognition. The Kinect sensor only detects colour and depth. However, if you have that information on your computer, you can perform a lot more, such as ”skeleton” tracking. While using Kinectron with Kinect for Windows we have used multiple feeds (frames) at the same time. We have used ‘Color’, ‘Depth’, ‘Raw Depth’, and ‘Skeleton (Tracked Bodies)’. All these files are in webm format, Skeleton also provides JSON files for 24 points located on the body. Dimensions of images are Color: 1920 x 1080 and Depth: 512 x 424.

3.0.1.2 Understanding Depth

The Kinect sensor provides the depth stream data as an extension of the depth field of view. With a viewing angle of 43 degrees vertically and 57 degrees horizontally, the sensor offers raw depth data in sixteen-bit gray-scale format. This is more than just a picture; it applies several algorithms to the data it gathers in order to provide more information than just an image 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.

3.0.1.3 Understanding Skeleton

The Kinect skeleton returns human body joints. The joints are numbered from 0 to 24. Color (x, y); depth (x, y); camera (x, y, z); and orientation (x, y, z, w) are the 11 attributes of each joint. The Kinect contains two cameras with various resolutions: a colour (RGB) camera and a depth camera. The colour camera has a resolution of 1920 by 1080. The depth camera has a resolution of 512 by 424. The colour coordinates in 2D and depth ones accommodate for this.

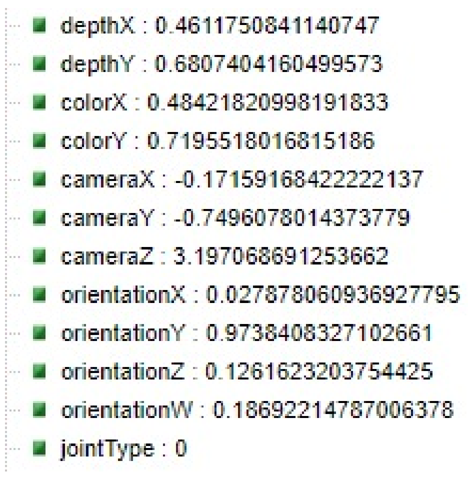
The **color coordinates** in 2D are the joint coordinates on the colour camera picture. A colour space point is a two-dimensional point on a colour picture. So a colour space position is a row/column location of a pixel on the picture, where x=0, y=0 corresponds to the top left pixel and x=1919, y=1079 refers to the rightmost in the bottom row. For colour coordinates, the sensor delivers a value belonging to [0,1] which is obtained by dividing the value with the resolution of sensor.

The **depth coordinates** in 2D are 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 colour coordinates, the sensor outputs value belonging to [0,1] for the depth camera as well. The Kinect's **camera coordinates** in 3D utilizes the IR beam to pinpoint the three-dimensional positions of the joints in the real world. The same are utilized in projects of three-dimensional joint placements. These are treated in a separate way than that of 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, x coordinate increases towards left
- With reference to the sensor's tilt, 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. Similarily, 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 object from the sensor, however, it can detect heights of up to five metres. The camera space's three values(x, y, z) are scaled on similar distance units to maintain proportions, and the unit used is meter.



```
depthX : 0.4611750841140747
depthY : 0.6807404160499573
colorX : 0.48421820998191833
colorY : 0.7195518016815186
cameraX : -0.17159168422222137
cameraY : -0.7496078014373779
cameraZ : 3.197068691253662
orientationX : 0.027878060936927795
orientationY : 0.9738408327102661
orientationZ : 0.1261623203754425
orientationW : 0.18692214787006378
jointType : 0
```

Figure 3: *Example of the coordinate space of a joint point.*

Joint	Index
Bottom of Spine	0
Middle of Spine	1
Collar	2
Face	3
Left Upper Arm	4
Left Arm joint	5
Left Carpus	6
Left Palm	7
Right Upper Arm	8
Right Arm Joint	9
Right Carpus	10
Right Hand	11
Left Pelvis	12
Left Patella	13
Left Talus	14
Left Foot	15
Right Pelvis	16
Right Patella	17
Right Talus	18
Right Foot	19
Shoulder to Spine	20
Left Palm Tip	21
Left Thumb	22
Right Palm Tip	23
Right Thumb	24

Table 1: *Joints corresponding to index values in the JSON file*

The Kinect contains two cameras with various resolutions: a colour (RGB) camera and a depth camera. The colour camera has a resolution of 1920 by 1080. The depth camera has a resolution of 512 by 424. The colour coordinates in 2D and depth ones accommodate for this. The Kinect's camera coordinates in 3D utilizes 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, and they must be translated to be helpful. [Quaternions] are 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.

3.0.2 Video Augmentation and Labelling

A deep learning model requires a large dataset for training. To accomodate this, we have applied augmentation on videos, by applying slight contrast changes, flipping frames, applying slight rotations or blurring frames. This helped in increasing the data almost 10 folds. Our videos are labelled into three classes, namely, Clear, Serve, and Smash. The augmented dataset is also available publicly.

A deep learning model generally works well when it has a huge amount of data. In general, the more data we have better will be the performance of the model.

Image augmentation is a technique of altering the existing data to create some more data for the model training process. In other words, it is the process of artificially expanding the available dataset for training a deep learning model.

Since all these images are generated from training data itself we don't have to collect them manually. This increases the training sample without going out and collecting this data. Note that, the label for all the images will be the same and that is of the original image which is used to generate them.

Now let us look at different image augmentation techniques

Image rotation one of the most commonly used augmentation techniques is image rotation. Even if you rotate the image, the information on the image remains the same. A cat is a cat, even if you see it from a different angle.

Hence, we can use this technique to increase the size of our training data by creating multiple images rotated at different angles.

Image Shifting The next image augmentation technique is image shifting. By shifting the images, we can change the position of the objects in the image and hence give more variety to the model. Which eventually can result in a more generalized model.

Image shift is a geometric transformation that maps the position of every object in the image to the new location of the final output image. So if an object is at position x,y in the original image, it gets shifted to new position X, Y in the new image. Where dx and dy are the respective shifts along with the different directions.

Image Flipping The next technique is Image flipping. Flipping can be considered as an extension of rotation. it allows us to flip the image in the Left-Right direction as well as the Up-Down direction.

Image Noising Another popularly used image augmentation technique is, Image Noising where we add noise to the image. This technique allows our model to learn how to separate the signal from the noise in the image. This also makes our model more robust to changes in the image

Image Blurring Let's understand one more augmentation technique, Image blurring. Images come from different sources and hence the quality of images will not be the same from each source. Some images might be of very high quality and others must be very bad. In such scenarios we can blur the original images, this will make our model more robust to the quality of the image being used in the test data.

3.0.3 Keypoint Detection

We have extracted 33 human body keypoints per frame by applying keypoint detection model on the augmented dataset of RGB videos. These keypoints are nothing but three dimensional coordinates of human joints.

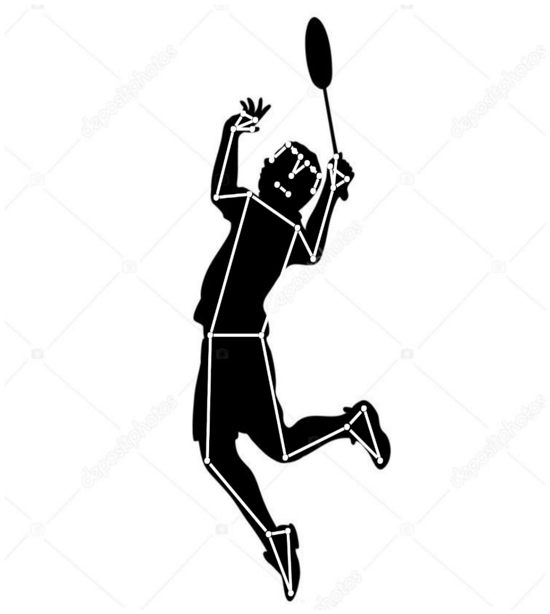


Figure 4: *Diagram showing Keypoints of Human Body Joints*

The whole frame has a lot of details, which may not correspond to the details needed to predict the type of shot played. This may even create hindrances in classification, which may result in bad accuracy. Also, using the whole video instead of extracting key points, will require much extra computation power, which may not be feasible. For these reasons, we chose to extract key points.

3.0.4 Classification Model

We have built a time-based sequential model to classify these badminton strokes into pre-defined classes. Our dataset is divided into two parts, 90 percent for the training set, and 10 percent for the test set.

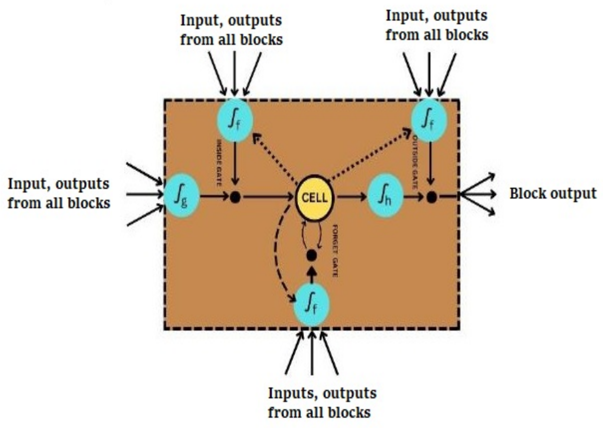

Input, outputs from all blocks
Input, outputs from all blocks
Input, outputs from all blocks
INSIDE GATE
OUTSIDE GATE
CELL
FORGET GATE
Block output
Inputs, outputs from all blocks

Figure 5: *Block diagram for LSTM unit*

We have used LSTM layers in our model, to get a hold of important features that may be present during any part of the stroke play. During model training, to further increase the training data, we have used a 5-fold cross-validation method. This serves the purpose of using the whole data in training, as well as in validation in a ratio of 80:20.

3.0.5 Evaluation Metrices

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

$$PrecisionforLabelA = \frac{TruePositivesoflabelA}{TotalpredictedinLabelA}$$

$$RecallforLabelA = \frac{TruePositivesoflabelA}{TotaltestitemswithlabelA}$$

$$AccuracyofModel = \frac{\sum_{c\in A} TruePositives(c)}{TotalItemsintestset}$$

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

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

$$SupportoflabelA = TotalitemswithlabelAintestset$$

3.1 Real Time Testing

We further tested our model by predicting the count of each class of sot played by the player near the camera in three different badminton videos taken from the official channel of the Olympics. The flow diagram of the process is shown below:

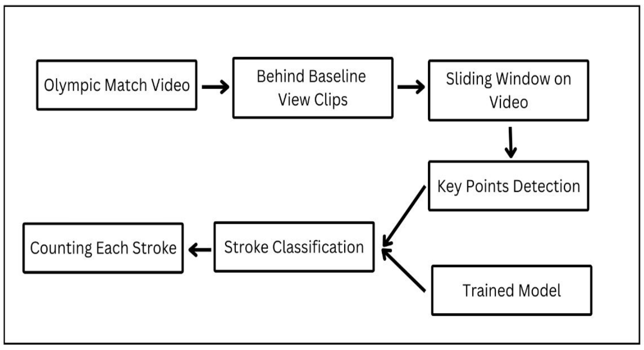


Figure 6: *Flow Diagram of real-time evaluation*

3.1.1 Highlights Subtraction

As we are only concerned with the real-time match video, which is recorded from behind the baseline view, we don't need to consider extra frames, focusing on the player, audience, and highlights of the game. For this purpose, we have taken out only those frames which are recorded from behind the baseline, by calculating the similarity of all frames, with a frame shot behind the baseline.

3.1.2 Sliding Window

After obtaining the cropped video, we run a sliding window on the clip. The window is shifted 9 frames at a time, and every time 15 alternating frames are taken and their key points are generated, in a similar fashion as explained before.

3.1.3 Stroke Classification and Counting

Supplying the key points as input data, we predict the label of the shot in that clip, using our pre-trained model. We then obtain the count of shots belonging to each class, as predicted by our model, and match the count with the actual count done manually.

3.2 Long Short Term Memory

The cell state is crucial for LSTMs. The cell state has some similarities to a conveyor belt. It moves down the entire chain immediately, with just a few minor linear interactions. It is quite simple for unaltered information to keep moving along it. The LSTM can change the state of the cell by adding or withdrawing information, which is meticulously regulated by gates.

Gates allow for the totally optional transmission of information. Both a layer of sigmoid neural networks 6 and a pointwise multiplication procedure make up these structures. The sigmoid layer produces integers between zero and one that represent the amount of each component that should be let to pass. Each block contains an input gate, a forget gate, and an output gate. Information is stored in the LTSM memory cell to be remembered over time. It has a self-loop that is modulated by a forget gate that ranges from 0 to 1, making the weight of the self-loop exactly equal to the value of the forget gate. It also features a linear activation function. The forget gate calculates a linear function of its input, then an activation function for logistics. The output is between 0 and 1.

The memory cell remembers its previous value when it is on (value = 1); else, it would be forgotten. The input gate allows the block to communicate with other blocks and receive input from them. They are combined and put through a tanh activation function, which has a range of -1 to 1. The memory cell value will be processed by another tanh activation function at the output, which may or may not transmit the value to the remainder of the network. Like the input and forget gates, the output gate modulates such a link. A rectified linear activation unit, or ReLU for short, is a node or component that implements this activation function

Chapter 4

Simulation and Results

For the purpose of collecting the dataset, we used Microsoft Xbox Kinect One Sensor, which provided us with RGB video, a depth heat map, and skeleton points of the player. After augmenting our videos, we used MediaPipe Keypoints Holistics Model to get the key points in the augmented dataset.

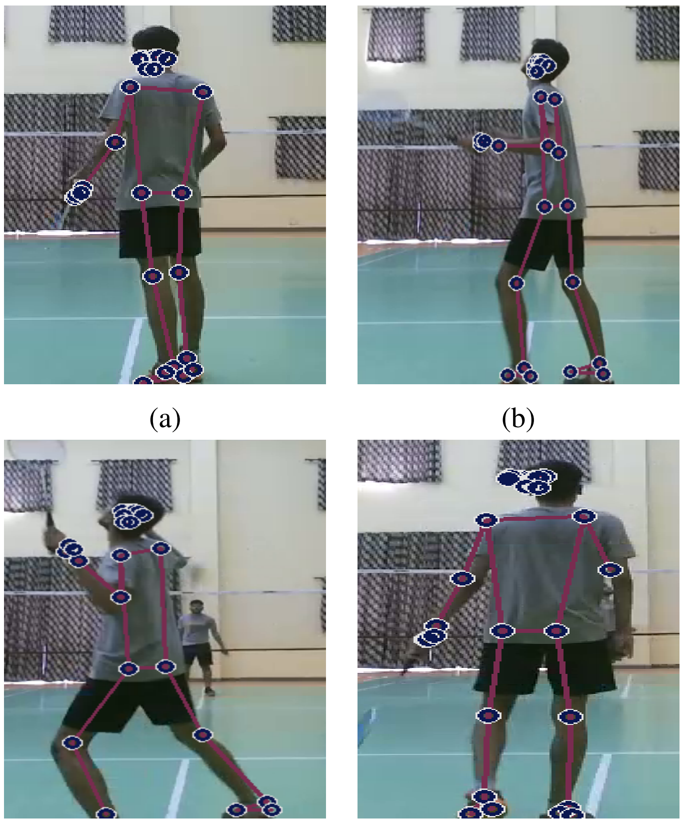


Figure 7: *Images(a-d) Shows plotted keypoints on a video of Clear shot*

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, and those videos which exceeded 15 frames generally contain some clip after the shot is played. We applied post-padding to those videos which have frames of less than 15.

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 Params: 2,56,435

Table 1: *Model Architecture*

After that, these keypoint sequences are fed to an LSTM(Long Short Term Memory) model, to classify our strokes into three classes, namely, Clear, Serve, and Smash. The model uses Adam Optimizer for calculating the step of gradient descent. The model architecture is shown in table 1. The model is trained with 5-fold cross-validation, and its performance is tested on a test set, with 10 percent of the total data. The final scores of the evaluation are shown in table 2.

	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
accuracy			0.89	552

Table 2: *Classification Report*

The confusion matrix obtained by evaluating the model on test set is shown in figure 7.

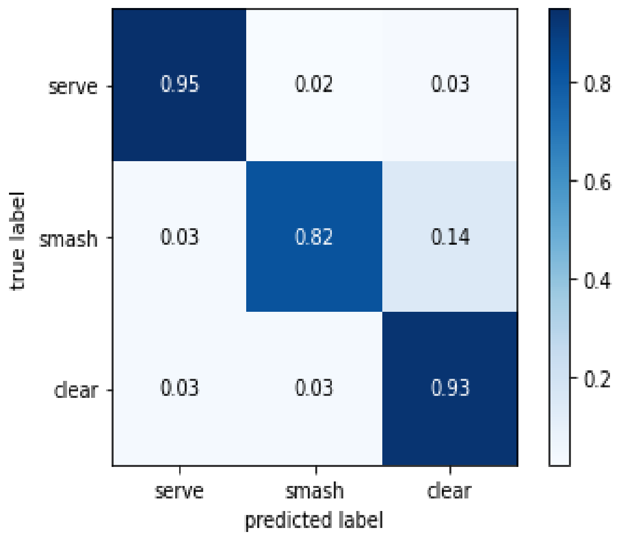


Figure 8: *Confusion matrix*

The count of each class as obtained by running the model on a real-time professional match is shown in the table below.

Match Index	Serve	Smash	Clear
1	42 / 2390	43 / 33	226 / 362
2	36 / 2818	22 / 10	102 / 261
3	58 / 4207	99 / 12	324 / 640

Table 3: *Strokes Count Per Class (Manual/Predicted)*

The results are obtained on 3 professional match shows, a large number of clips classified as serve. This anomaly arises because of the missing class ”others”. In all the clips in which the player is not playing any shot, his movements are minimal, similar to that of the serve shot. Hence, most of these clips are also classified as serve. Also, as we are shifting only 9 frames at a time, so as to not miss any shot, there are chances that one shot’s beginning and the ending come in two consecutive clips, which correspond to the increased number of smash and clear shots. Also, there are many other types of shots, which are played quite similarly to our defined shots but are classified in one of our defined classes, due to the absence of the ”others” class.

Chapter 5

Conclusions and Future Work

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 percent accuracy on the test set. Further, we have applied our trained model to real-life professional match videos and obtained the count of each class of shot played in the match as predicted by our model.

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.

Bibliography

[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 (MM-Sports '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.