Plagiarism Report

Originality Assessment

28%

Overall Similarity

Date: Oct 16, 2023
Matches: 2121 / 5068 words
Sources: 21

Remarks: High similarity detected, you must improve the document.

Verify Report:

Detection of Cricket Shots Using Deep Learning

Manan Pruthi1, Ashish Katyal1, Sanyam1, Rishabh Semwal1, Vijay Kumar1

1Department of Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering,

New Delhi-110063, India

Abstract-- Classifying the different types of bat shots played in cricket has always been a challenging task in the field of cricket indexing. Identifying the type of shot a batsman played during a match is a crucial aspect that has not been thoroughly studied. This information can be used for context-based advertisements for cricket viewers, creating sensor-based commentary systems, and coaching assistants. However, manually identifying the different shots from video frames is difficult due to the similarity between them. This project presents a new approach for recognizing and categorizing different cricket shots by using state-of-the-art techniques such as saliency and optical flow to capture both static and dynamic information, and Long Short Term Memory (LSTM) for representation extraction. Additionally, a new dataset of

120 videos has been introduced to evaluate the performance of the model, with 4 classes of shots each having 30 videos. The model achieved an accuracy of 83.34% for the four classes of cricket shots.

Index Terms--Cricket, Convolutional Neural Network, LSTM, MediaPipe

I. INTRODUCTION

Cricket is a game played with a bat and ball and various technologies are used to visualize and coach it. However, recent research has not yet produced satisfactory results for detecting shots in the game. The challenge lies in extracting features from video footage captured from different cameras. Some methods utilize information such as field position, camera position, boundary limits,

commentary, and texts. However, machines 9 are unable to visualize smoothly like humans do. The issues involved in detecting shots include recognizing the movements of the human body parts, the cricket bat, and the ball swing. Machines can detect human

body parts in images, but tracking and estimating the movement of these body parts is difficult using a single video. Semantic analysis for cricket broadcasting video requires a large dataset, but its accuracy level is still not satisfactory. Action classification based on templates is used to distinguish between different sport activities, but it is challenging to classify cricket shots due to the obscuration of the bat and body parts. This study proposes a unique approach to classify cricket shots by using motion vector detection for each frame using optical flow and transforming the vectors into angles.

The recent surge in multimedia content, such as images and videos, has led to a boom in research across various disciplines, including sports and athletics where a vast amount of data is available for analysis. Cricket, as a globally popular sport, is widely broadcasted, providing a variety of camera shots and events for analysis and estimation. Classifying different types of cricket shots is a difficult but crucial aspect of indexing cricket games. There are various standard cricket bat strokes such as Late Cut, Pull, Cover Drive, Leg Glance, etc., that are played during a match and sometimes with slight variations. In order to improve a batsman's game and minimize deviation from the standard technique, shot detection algorithms are highly valuable. The difficulty in recognizing different cricket shots lies in extracting unique features and cues from a limited number of data frames, which calls for the development of automated systems that can effectively detect cricket shots from real-time videos.

Fig 1: Various cricket shots played during a match [15]

Classifying the type of bat stroke played in a cricket match is a difficult task. The success of a

shot is not always guaranteed, making it challenging to accurately determine the intended shot played. Factors like occlusion, the small size of the ball, and the fast pace of the game only add to the complexity. Sifting through the data to find only the relevant information is a time-consuming and challenging process, requiring a large data set.

1.1 Problems in Current Scenario

6 One of the major hurdles is detecting the movements and posture of body parts from just a single video. Furthermore, semantic analysis of cricket broadcasting video requires a big dataset, with accuracy being a concern. Distinguishing between different sports activities relies on template-based action classification which makes it hard to classify cricket shots because of the occlusion of the bat and body parts.

II.

OBJECTIVES

The task of categorizing different types of batting shots in a cricket game has been a challenging task in the indexing of cricket. Knowing the type of shot played by a batsman in a cricket match is an important and untapped aspect in this field. The goal is to introduce a new method for recognizing and classifying various bat shots played in cricket.

In this project, the aim is to identify the cricket shot a batsman is playing as it happens. Previously, other projects have succeeded in doing so with recorded videos, but this project focuses on using real-time videos captured from the front camera.

This breakthrough can lead to the creation of an automatic commentary system, a significant advancement for the sports industry. Additionally, it can be used to develop an automatic highlight generation system and provide a batsman with valuable statistics to improve their gameplay.

III. LITERATURE REVIEW

Recently, there has been an increase in research in the field of sports due to its commercial value and widespread popularity. Different methods have been developed to classify various actions in sports videos. Some of these methods have been used in the analysis of cricket videos, categorizing different types of shots played in the game.

Classification of eight cricket shots was performed using deep convolutional neural networks (CNNs) in [1]. A 2D CNN, 3D CNN, and an RNN with LSTM were utilized to categorize the shots. The classification was improved through transfer learning from a pre-trained VGG model. The highest accuracy of 90% was obtained with 800 shots, but the hook shot had the lowest accuracy at 85%. In

[2], a new method for classifying cricket shots was proposed using motion estimation. The classification was done based on eight different angle ranges.

The highest accuracy of 63.57% was recorded for the off drive, while the hook shot had the lowest accuracy of 53.32%. In [3], wearable technology was utilized to categorize cricket shots with the goal of creating a quality evaluation system for these shots. The classification was based on specific aspects such as foot position and shot direction using hierarchical representation. Five levels of grouping actions were determined using DT, SVM, and k-NN algorithms. The best classifiers per level gave a class-weighted F1 score of 88.30%. This work only used videos collected from a training session, not actual match videos. [4] presented a CNN approach to classify 6 different cricket shots based on a custom-developed dataset. [5] introduced a model to identify straight drive, cover drive, and lofted on drive for both left- and right-handed batsmen using a pretrained CNN and multiclass SVM. The model achieved 83.098% accuracy for right-handed shots and 65.186% accuracy for left-handed shots. Misclassifications occurred for some shots such as left cover drive and straight drive due to the limited dataset.

Several efforts have been made to identify different events in cricket videos [6,7]. Harikrishna et al. [6] presented a technique that divided cricket video into shots and used SVM to classify these shots into four events, including run, four, six, and out. The recorded accuracy was 87.8%, but it had low accuracy for the six events. Another study was conducted by Kolekar et al. [7]. They presented a hierarchical framework for identifying different events in cricket videos, including excitement detection, replay detection, field-view detection and classification, close-up detection and classification, and crowd classification. They mentioned that there was a problem with misidentification near the boundaries of the shots. Premaratne et al. [8] proposed a structured method for recognizing different events in a full cricket match. They used k-nearest-neighbor, sequential-minimal optimization, decision-tree, and naive Bayes classifiers for event detection. However, the authors noted the need for human interpretation to remove replay frames and highlights between deliveries.

Several studies were carried out for automatically extracting highlights from cricket videos [9-14]. In one study [9], video and audio features were combined. The researchers started by using rule-based induction to detect exciting audio clips, followed by a video summarization step using a decision tree. This method achieved an average accuracy of 90%. In [10], the authors suggested a hierarchical approach 20 for automatically generating highlights from cricket videos. They used association-rule mining 1 to extract semantic concepts and noted the importance of features such as detecting the umpire's gesture and estimating camera motion as future work. In [11], methods for summarizing cricket games based on video-shot detection, sound detection, and scoreboard recognition were proposed. In [12], a new method for identifying highlights in sports videos was presented. The approach used a color histogram and a histogram of oriented gradients. [13] introduced a unique summarization strategy that extracted boundary and wicket events based on bowling and score changes, and removed ads and replay events through calculation of the temporal derivative. In [14], a hierarchical approach was proposed for real-time extraction of highlights in sports videos, using a combination of motion and color features within a multi-spatial framework. Ringis et al. [15] proposed a method to automatically detect bowler run-up sequences (BRSs) in cricket to generate highlights, using the oriented fast and rotated brief (ORB) method. However, their proposed method has some limitations, such as the requirement of training at the start of the match, detection of players as key points only, no use of orientation parameters, and only training and detection of BRS views. Rafiq et al. proposed a method in [16] to generate video summarization in sports videos using transfer learning. The approach used a CNN-based Alexnet model and achieved an accuracy of 99.26% when tested on a smaller dataset of only cricket scenes.

So far, only a few experiments involving sports other than cricket have been conducted. Steels et al. [17] conducted experiments in sports other than cricket and proposed a method that uses a convolutional neural network and accelerometer data to recognize 9 different badminton actions. The accuracy of the proposed method was lower compared to the average accuracy, with a weighted accuracy of 93% and 96% for sensor placements in the wrist and upper arm, respectively. Rangasamy et al. [18] proposed a model that used the pre-trained VGG16 network to

classify four types of hockey activities: free hit, goal, penalty corner, and long corner. After 300 epochs, the model achieved an accuracy of 98%, but it struggled to distinguish between a free hit and a long corner because they had similar visual patterns with regards to player positioning and appearance. In [19], Junjun et al. put forth a method to identify three basketball actions: shoot, pass, and catch. The model utilized an adaptive multi- level classification technique and achieved an accuracy of 92.3%. However, this study was limited in scope, 4 covering only a small number of examples. In [20], a method for recognizing fine-grained video actions in basketball games was proposed by Gu et al. The study focused on 3 broad actions, dribbling, passing, and shooting, which were further divided into 26 fine-grained actions. However, 1 the average precision of subclasses within the broad categories varied significantly due to their unique characteristics.

IV. METHODOLOGY

4.1 Dataset Description

Since there was no publicly available dataset for individual cricket shots, a dataset was created manually by recording various shots in real time using OpenCV and a front camera. The data set was constructed to meet the desired specifications. A series of cricket shots were recorded with a front-facing camera, which were then divided into four shot categories, each with 30 video clips. The final data set 4 consisted of around a total of 120 cricket shot recordings. The video clip distribution for each category is described in Table

3.1. The experiments were conducted using 10 an AMD Ryzen 5

Hexacore 5500U processor and NVIDIA GeForce GTX 1650 GPU, and were run using Jupyter Notebook as the software.

TABLE 1: Total Data Set Count Under Each Class Label

CRICKET SHOT	NO. OF VIDEOS
Pre-Stance	30
Stance	30
Straight Drive	30
Pull Shot	30

Figure 2: Examples from the dataset, (a) Straight-drive, (b) Stance

4.2 MediaPipe

Object detection is 3 a widely used and popular application in the field of computer vision. Many object detection models are applied globally for various specific purposes. Many of these models are designed to be standalone solutions to individual computer vision problems and are fixed to a particular use case. Combining all 5 of these tasks into a single

end-to-end solution, in real-time, is basically the job of MediaPipe.

MediaPipe is an open-source, cross-platform machine learning framework that allows for the creation of complex, multi-modal machine learning pipelines. It can be used to develop advanced machine learning models for tasks such as face detection, multi-hand tracking, object detection and tracking, and more. MediaPipe acts as a mediator for implementing models on any platform, freeing developers to focus on experimenting with models rather than the underlying system.

4.2.1 MediaPipe Holistic

Mediapipe Holistic is a pipeline that has been optimized to track faces, hands, and poses all at once, which is referred to as "holistic tracking." This pipeline enables detection of hand and body poses along with facial landmarks. It is mainly used for detecting faces and hands and extracting key points for use 3 in computer vision models.

4.2.2 Detect face, hand, pose 5 landmarks and extract keypoints

The MediaPipe Holistic pipeline brings together optimized models for human pose, face, and hand detection, allowing for the simultaneous tracking of hand and body poses along with face landmarks. This pipeline 3 can be used for various applications that require human pose estimation from video, such as measuring physical exercises, recognizing sign language, and controlling full-body gestures. These capabilities can be applied in fields such as yoga, dance, and fitness.

16 The MediaPipe Pose pipeline uses machine learning to accurately track body pose, including 33 3D landmarks that consist of x, y, z and visibility coordinates. 8 The MediaPipe Face Mesh, on the other hand, is capable of estimating 468 3D face landmarks in real-time. The Face Landmark Model performs face landmark detection from a

single camera and expresses the X- and Y- coordinates as normalized screen coordinates, with the Z coordinate being relative and scaled relative to the X coordinate based on the weak perspective projection camera model.

MediaPipe Hands is a machine learning-based solution for tracking hands and fingers with high precision. It uses ML to identify the 15 21 3D landmarks of a hand from just one frame, and each hand is represented as a list of 21 landmarks, each with x, y, and z coordinates.

Fig 3: Model Architecture

A video is divided into 30 frames, and 9 the coordinates of 1662 points are calculated for each frame. The 30 frames from one video are then sent to a model for prediction of 1 a cricket shot action (Pre-Stance, Stance, Straight Drive, Pull Shot) based on these 1662 points. The input for 6 the model is 30 sets of 30 frames each, with 1662 points per frame, for a total of 30*30*1662 per action.

4.3 Pre-processing of Data

Initially, a large NumPy array was created to hold all the data, which is a common first step in the pre-processing of imported data before building a model. So, we have 4 actions labelled as:

[Pre-Stance : 0, Stance : 1, 1 Straight Drive : 2, Pull Shot : 3]

The dataset consisted of 120 videos of cricket shots, each action contributing 30 videos. Each video had 30 frames and each frame had 1662 key-points, all of which were stored in a single NumPy array.

Therefore, the final result was an array of 120 arrays, each with 30 frames, and each frame having 1662 values which represents the key points. This resulted in an array shape of (120, 30, 1662).

The label array, containing 120 values, with 30 for each action, was created. The True Categorical function was utilized 8 to convert the label array into a binary class matrix. 1 As a result, the Pre-Stance was depicted by [1,0,0,0] instead of 0, Stance was represented by [0,1,0,0] instead of 1, Straight Drive was represented through [0,0,1,0] instead of 2, and Pull Shot was shown via [0,0,0,1] instead of 3.

The dataset was divided 21 into training and testing sets using the train_test_split function. The training partition consists of 95% of the total data, while the test partition consists of 5% of the data.

4.4 LSTM Model

LSTMs 7 are a type of Recurrent Neural Network that is designed to handle sequence prediction problems. They work by using the output from a previous step as input for the current step, allowing them to learn the order dependencies in a sequence. LSTMs were developed by Hochreiter and Schmidhuber. The solution tackles the problem of long-term dependency in RNNs, where the network struggles to recall information from its memory but performs better in predicting based on current input. RNNs struggle to predict words from memory as the gap length increases, whereas LSTMs can store information for a longer time and are therefore more effective for time-series data processing, prediction, and classification tasks. LSTM is distinct from traditional feed-forward neural networks 3 due to its feedback connections, which allow it to process not only individual data points

(e.g. images) but also continuous sequences of data (such as audio or video). This makes LSTM suitable for tasks such as unbroken handwriting recognition and speech recognition.

4.4.1 2 Structure of LSTM

The LSTM is composed of four neural networks and many memory units called cells arranged in a chain structure. A standard LSTM unit comprises a cell, 7 an input gate, an output gate, and a forget gate. The input, output, and information flow into and out of the cell is regulated by the three gates, allowing the cell to retain values over extended periods of time. The LSTM model is well-suited to classify, examine, and forecast time-series data of indefinite length.

The cells retain information, while the gates control memory. 2 There are three entrances:

Input Gate: The LSTM decides which of the input values should affect the memory, using the sigmoid function to determine whether to pass on 0 or 1 values and the tanh function to assign significance to the input data on a range of - 1 to 1.

Forget Gate: The LSTM determines which details 7 to discard from the memory block, as decided by the sigmoid function. For each element in 14 the cell state at time C_{t-1}, it considers the previous state (at time h_{t-1}) and the current input (at time X_t), and outputs a value between 0 (to discard) and 1 (to keep).

2 Output Gate: The output is determined by the block's input and memory, with the sigmoid function controlling which 0 or

1 values are permitted to pass and the tanh function regulating which values can pass through 0 or 1. The tanh function also assigns importance to the input data on a range of -1 to 1 and multiplies this with the output from the sigmoid function.

The recurrent neural network employs long short-term memory (LSTM) blocks to give context to how the software handles inputs and produces outputs. By using a design inspired by short-term memory processes to construct long- term memory, the unit is known as a long short-term memory block. In natural language processing, these systems are widely utilized.

For instance, the recurrent neural network utilizes long short- term memory blocks to examine a single word or sound in the context of the others in a sequence, as the memory can assist in the filtering and classification of specific types of information. Overall, LSTM is a well-established and widely used concept in the design of recurrent neural networks.

Fig 4: Structure of LSTM [11]

4.4.2 LSTM Networks

Recurrent neural networks are composed of a series of repeating neural network modules. In conventional RNNs, this repeating module has a straightforward structure, such as a single tanh layer. The output from one time step becomes the
input for the next, which is referred to as being "recurrent." At each element in the sequence, the model takes into account not only the current input but also its knowledge of previous inputs. The cell state, depicted by the horizontal line at the top of the diagram, is a critical component of LSTMs.

The cell state operates similarly to a conveyor belt in some ways, with only a few small linear interactions along its length. Data can easily flow down it without being changed. The LSTM can choose to remove or add information to the cell state, which is precisely regulated by structures known as gates. Gates are a mechanism for selectively letting information 12 pass through and are comprised of a sigmoid neural network layer and a pointwise multiplication operation.

2 The sigmoid layer generates numbers between zero and one, showing how much of each component should be permitted to pass. A value of zero means "nothing" should be allowed, while a value of one means "everything" should be allowed. An LSTM has three of these gates to control and manage the cell state.

4.4.3 LSTM Cycle

The LSTM process consists of four phases. The forget gate identifies information to be discarded from the previous step. The input gate and tanh identify new information for updating the cell state. The information from the previous two gates is then used to update the cell state.

The LSTM cycle is comprised of four steps, including the identification of information to be forgotten using the forget gate, and the update of the cell state using the input gate and tanh. The output of an LSTM cell is passed through a dense layer and then processed through the softmax activation function in the output stage.

4.4.4 Bidirectional LSTMs

Bidirectional LSTMs are an improvement on traditional LSTMs, which is often discussed. In a Bidirectional Recurrent Neural Network (BRNN), the same output layer is connected to two independent recurrent networks that process each training sequence forwards and backwards. This provides the BRNN with comprehensive sequential knowledge about all points before and after each point in a sequence. With BRNN, there is no need to specify a time window size or goal delay as it has the freedom to use as much or as little context as needed.

Traditional RNNs have the limitation of only being able to utilize prior context. To overcome this, Bidirectional RNNs (BRNNs) process data in both directions using two hidden layers that both feed into the same output layer. By combining LSTMs with BRNNs, a bidirectional LSTM is created that can access context from both input directions over a long range.

4.4.5 Proposed LSTM model

In this project, we have constructed an LSTM model with 3 LSTM layers followed by 3 Dense layers, 9 as shown in the accompanying figure.

FIG 5: LSTM MODEL

V. RESULTS

3 To determine the effectiveness of the LSTM model in this project, the performance was measured using categorical accuracy. Graphs for both categorical accuracy and categorical cross entropy were also analyzed.

Since 1 there was no publicly available dataset for individual cricket shots, short video clips of

cricket shots were made to create a dataset. The dataset was then created manually to meet the specific needs, due to the lack of sufficient data. 3 In the end, the dataset was comprised of around 120 cricket shot recordings, divided into 4 categories, with 30 recordings for each shot type. These experiments were conducted on a system 10 with an AMD Ryzen 5 Hexacore 5500U processor and an NVIDIA GeForce GTX 1650 GPU, using Jupyter Notebook as the software.

1 The dataset was then divided into training and testing sets using the train_test_split function, with the training set consisting of 95% of the total data and the test set consisting of 5% of the data.

The label array was created with 120 values, each representing one of the four shot actions (Pre-Stance, Stance, Straight Drive, Pull Shot). The true categorical function was applied to convert the label array into a binary class matrix, where Pre- Stance was represented by [1,0,0,0], Stance by [0,1,0,0], Straight Drive by [0,0,1,0], and Pull Shot by [0,0,0,1].

In our project, we employed the LSTM algorithm and built a model with three LSTM layers followed by three dense layers. The model was trained on shots made with both the left and right hand.

The model was evaluated using real-time video input from a camera and video clips from international cricket matches.

Fig 6: Pull Shot detection

Fig 7: Straight Drive detection

Our model achieved an overall accuracy of 83.33%, surpassing the results of many previous models.
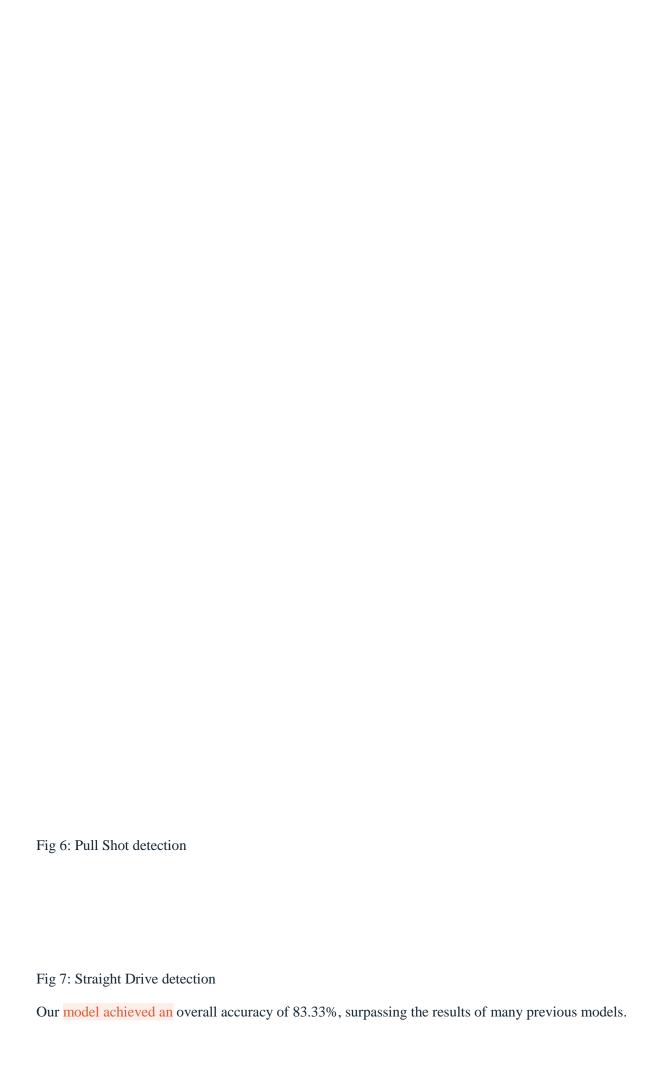
Fig 8: Categorical Accuracy v/s Epochs

Fig 9: Categorical Cross-entropy v/s epochs

VI.

CONCLUSION

The project introduces a new approach for detecting various shots in cricket. The framework proposed utilizes a 18 Long Short Term Memory (LSTM) Neural Network to extract information from the three components of the model. 9 The proposed method performed better than prior state-of-the-art models and achieved a promising detection accuracy. Additionally, a new data set of 120 videos for 4 classes of cricket shots was introduced. A more sophisticated 6 model can be highly beneficial for professional cricket training centers, as it can accurately recognize skills. 1 The cricket shot detection algorithm can be utilized for visualization and coaching, and a more advanced model could be applied in automated commentary systems. 9 Future work could focus on expanding the recognition classes and incorporating more specific features for each defined shot.

With the increasing popularity of shorter format cricket matches like T20, more unconventional shots are being played. Our future work aims to incorporate 1 these unorthodox shots. Furthermore, classifying umpire actions and boundary events will greatly improve the accuracy of automatic highlight extractions. These actions will also be addressed in future work. Our goal is to expand the dataset and develop a custom model with higher accuracy. Another challenge we plan to tackle is detecting keyframes from a video, as this would allow us to work with fewer frames and parameters. Some experiments combining acoustic and image features will be conducted. It would be intriguing to combine traditional machine learning methods like SVM, SIFT, and visual bag of words with deep learning algorithms to study the performance of the model. The proposed architecture can be applied to solve various video classification problems. However, GPU memory limitations hinder the combination of complex models and large datasets.

VII. REFERENCES

[1] Khan, M.Z.; Hassan, M.A.; Farooq, A.; Khan, M.U.G. Deep CNN Based Data-Driven Recognition of Cricket Batting Shots. In Proceedings of the International Conference onApplied and Engineering Mathematics (ICAEM), Taxila, Pakistan, 4–5 September 2018; pp.67–71. [CrossRef]

[2] Karmaker, D.; Chowdhury, A.; Miah, M.; Imran, M.; Rahman, M. Cricket shot classificationusing motion vector. In Proceedings of the 2nd International Conference on Computing Technology and Information Management (ICCTIM), Johor, Malaysia, 21–23 April 2015; pp. 125–129. [CrossRef]

[3] Khan, A.; Nicholson, J.; Plötz, T. Activity recognition for quality assessment of batting shots in cricket using a hierarchical representation. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2017, 1, 1–31. [CrossRef]

[4] Foysal, M.F.A.; Islam, M.S.; Karim, A.; Neehal, N. Shot- Net: A convolutional neural network for classifying different cricket shots. In International Conference on Recent Trends in Image Processing and Pattern Recognition; Springer: Berlin, Germany, 2018; pp. 111–120.

[5] Semwal, A.; Mishra, D.; Raj, V.; Sharma, J.; Mittal, A. Cricket shot detection from videos. In Proceedings of the 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 10– 12 July 2018; pp. 1–6. [CrossRef] Sensors 2021, 21, 2846 19 of 19

[6] Harikrishna, N.; Satheesh, S.; Sriram, S.D.; Easwarakumar, K. Temporal classification of events in cricket videos. In Proceedings of the Seventeenth National Conference on Communications (NCC), Bangalore, India, 28– 30 January 2011; pp. 1–5. [CrossRef]

[7] Kolekar, M.H.; Palaniappan, K.; Sengupta, S. Semantic event detection and classification in cricket video sequence. In

Proceedings of the Sixth Indian Conference on Computer Vision, Graphics & Image Processing, Bhubaneswar, India, 16–19 December 2008; pp. 382–389. [CrossRef]

[8] Premaratne, S.; Jayaratne, K. Structural approach for event resolution in cricket videos. In Proceedings of the International Conference on Video and Image Processing, Singapore, 27–29 December 2017; pp. 161–166. [CrossRef]

[9] Javed, A.; Bajwa, K.B.; Malik, H.; Irtaza, A.; Mahmood, M.T. A hybrid approach for summarization of cricket videos. In Proceedings of the IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Korea, 26–28 October 2016; pp. 1–4. [CrossRef]

[10] Kolekar, M.H.; Sengupta, S. Semantic concept mining in cricket videos for automated highlight generation. Multimed. Tools Appl. 2010, 47, 545–579. [CrossRef]

[11] Bhalla, A.; Ahuja, A.; Pant, P.; Mittal, A. A Multimodal Approach for Automatic Cricket Video Summarization. In Proceedings of the 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, Uttar Pradesh, India, 7–8 March 2019; pp. 146–150. [CrossRef]

[12] Tang, H.; Kwatra, V.; Sargin, M.E.; Gargi, U. Detecting highlights in sports videos: Cricket as a test case. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Barcelona, Spain, 11–15 July 2011; pp. 1–6. [CrossRef]

[13] Kumar, Y.S.; Gupta, S.K.; Kiran, B.R.; Ramakrishnan, K.; Bhattacharyya, C. Automatic summarization of broadcast cricket videos. In Proceedings of the IEEE 15th International Symposium on Consumer Electronics (ISCE), Singapore, 14– 17 June 2011; pp. 222–225. [CrossRef]

[14] Ramsaran, M.; Pooransingh, A.; Singh, A. Automated Highlight Generation from Cricket

Broadcast Video. In Proceedings of the 8th International Conference on Computational Intelligence and Communication Networks (CICN), Tehri, India, 23–25 December 2016; pp. 251–255. [CrossRef]

[15] Ringis, D.; Pooransingh, A. Automated highlight generation from cricket broadcasts using ORB. In Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, Canada, 24–26 August 2015; pp. 58–63. [CrossRef]

[16] Rafiq, M.; Rafiq, G.; Agyeman, R.; Jin, S.I.; Choi, G.S. Scene classification for sports video summarization using transfer learning. Sensors 2020, 20, 1702. [CrossRef] [PubMed]

[17] Steels, T.; Van Herbruggen, B.; Fontaine, J.; De Pessemier, T.; Plets, D.; De Poorter, E. Badminton Activity Recognition Using Accelerometer Data. Sensors 2020, 20, 4685. [CrossRef] [PubMed]

[18] Rangasamy, K.; As'ari, M.A.; Rahmad, N.A.; Ghazali, N.F. Hockey activity recognition using a pre-trained deep learning model. ICT Express 2020, 6, 170–174. [CrossRef]

[19] Junjun, G. Basketball action recognition based on FPGA and particle image. Microprocess. Microsyst. 2021, 80, 103334. [CrossRef]

[20] Gu, X.; Xue, X.; Wang, F. Fine-Grained Action Recognition on a Novel Basketball Dataset. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2563–2567

Sources

1
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8072636/
INTERNET
10%

2
https://www.analyticsvidhya.com/blog/2022/03/an-overview-on-long-short-term-memory-lstm/#:~:text=It
addressed the issue of RNN long-term dependency,,keep information for a long time by default.
INTERNET
8%

3
https://www.mdpi.com/1424-8220/23/9/4258
INTERNET
2%

4
https://www.researchgate.net/publication/374673576_Cricket_Gesture_Detection_Using_LSTM_Media_Pipe_And_CN
N
INTERNET
1%

5
https://www.geeksforgeeks.org/python-facial-and-hand-recognition-using-mediapipe-holistic/#:~:text=Combining
several of these tasks into a single,building complex and multimodal applied machine learning pipelines.
INTERNET
1%

6
https://www.nature.com/articles/s41598-022-05966-6
INTERNET
1%

7
https://en.wikipedia.org/wiki/Long_short-term_memory
INTERNET
1%

8
https://github.com/google/mediapipe/wiki/MediaPipe-Face-Mesh
INTERNET
1%

9
https://link.springer.com/article/10.1007/s00521-021-06322-x
INTERNET
<1%

10
https://www.flipkart.com/acer-aspire-7-ryzen-5-hexa-core-5500u-8-gb-512-gb-ssd-windows-10-home-4-graphics-
nvidia-geforce-gtx-1650-a715-42g-gaming-laptop/p/itm4385fddc2c72c
INTERNET
<1%

11
https://www.researchgate.net/profile/Azm-
Chowdhury/publication/283090676_Cricket_Shot_Classification_Using_Motion_Vector/links/56d9e19508aee1aa5f82
9686/Cricket-Shot-Classification-Using-Motion-Vector.pdf
INTERNET
<1%

12
https://www.mdpi.com/1424-8220/21/13/4592
INTERNET
<1%

13
https://ijcrt.org/papers/IJCRT23A5478.pdf
INTERNET
<1%

14
https://www.chegg.com/homework-help/questions-and-answers/lstm-model-explain-exactly-cell-state-updated-
ct-1-ct-using-previous-state-ht-1-current-in-q83102247
INTERNET