

A SYSTEM FOR AUTOMATICALLY DETECTING NO BALL IN CRICKET THROUGH MACHINE LEARNING TECHNIQUES

FIELD OF THE INVENTION

- 5 The present disclosure is related to artificial intelligence and machine learning techniques and a system for automated detection of No-Ball in the game of cricket.

BACKGROUND OF THE INVENTION

Cricket is the second most viewed sport in the world which is played between two opposing teams by bat and ball on a field at the center of the pitch i.e.; 22-yard (20.12 meter) with a wicket at each
10 end called popping crease.

In the game of cricket, one team has to ball, and the second has to bat. There are some parameters set for the bowlers to throw the ball. No ball is an illegal action in cricket, which is decided by the Umpires. Sometimes, it is missed by the Umpire even in world-class tournaments which make the opponent's team unhappy. Computer vision and Artificial Intelligence models are used in different
15 sports to analyze the matches, measure the player KPIs (key performance indicators), and also to detect fouls, helping the referee to make the match a fair one.

In view of the foregoing, the present disclosure relates to the system for detecting delivery action in cricket sports, tagging the video, and specifying the action type, especially detecting the No-Ball foul during the course of the match. The Umpire is notified with a notification system, right
20 at the moment that a No-Ball happens.

The system consists of multiple neural networks for detecting the No-Ball with the help of popping crease detection module, delivery action detection module, bowler detection module detecting the bowler's front foot heel at landing frame further, the system measures the distance between the popping crease and the bowler's front foot heel at the landing frame and if it is a No-Ball, a
25 notification system sends a notification message to the Umpire.

DESCRIPTION OF RELATED PRIOR ARTS

The prior art US 20210256265 A1 discloses a computing system that retrieves ball-by-ball data for a plurality of sporting events. The computing system generates a trained neural network based on ball-by-ball data supplemented with ball-by-ball data with ball-by-ball match context features and personalized embeddings based on a batsman and a bowler for each delivery.

- 5 The prior art US 20200054928 A1 discloses a system and method for analyzing human body movements relevant to various sports. The system requires connecting a multiplicity of sensors at specified locations of the human body. Raw data is gathered from the sensors and processed to derive various informative parameters. The informative parameters are displayed to the pupil with descriptive feedback. The system also features communication of the raw data and informative
10 parameters to external entities through a computer network.

The prior arts US 20130274904 A1, US 20130274587 A1, and US 10286278 B1 disclose Wearable Athletic Activity Monitoring Systems.

OBJECT OF THE INVENTION

- 15 It is an object of the present invention to provide artificial intelligence and machine learning techniques and a system for automatically detecting No-Ball in the game of cricket. The use of the automatic No-Ball detection system drops human errors on No-Ball detection and makes the game a fair one.

It is another object of the invention to provide the No-Ball decision instantly with more accuracy and save time.

20 SUMMARY OF THE INVENTION

Sometimes it is not easy for the Umpire to detect the No-Ball since it happens quickly and it is hard to see. In cricket tournaments, it is usual that some No-Balls are missed by the Umpires, and it makes the competing teams unhappy. The use of an automatic No-Ball detection system drops human errors on No-Ball detection and makes the game a fair one.

- 25 The system for automated detection of No-Ball consists of multiple neural network modules for detecting the popping crease, detection of delivery action, detection of bowler, detection of bowler front foot landing frame, and detecting the bowler's front foot, heel at landing frame. The system measures the distance between the popping crease and the bowler's front foot, heel at the landing

frame and if it is a No-Ball, a notification system sends a notification message to the Umpire. The Umpire is notified with the notification system, right at the moment that a No-Ball happens.

BRIEF DESCRIPTION OF THE INVENTION

The system for automatically detecting No-Ball through machine learning techniques comprising
5 of a data stream module for providing raw data or sequence of frames from a live broadcast data or a recorded video file. The sequences of frames are received from a live broadcast or a recorded video file, they could be in any number of frame rates, of any resolution, compressed in any form of compression, or provided as raw data. This module converts the stream of data to a sequence of discrete frames and feeds them to the next module i.e. line detection module.

10 The main purpose of the line detecting module is to detect popping-crease on the cricket pitch as mentioned in figure 2. Some portion of the bowler's front foot at the landing moment should stay behind this line unless it is a no-ball as mentioned in figure 4. The line deduction module works with the help of an input model that takes one RGB image (Red-Green-Blue, primary colours image) with the size of 224x224x3mm from the data stream module and adds a convolution layer,
15 17 bottleneck residual blocks that are constructed from convolution layers, Dwise convolution, a residual connection, and a dropout layer to have regularization effect. Two separable convolution layers are added, the first layer has a kernel size of 5 and an activation function of relu and, the second layer has a kernel size of 3 and an activation function of sigmoid, so the output of this layer has four values which corresponds to coordinates of two points. The line detecting module has an
20 output model. It is two ends of a popping crease. This makes it possible to build the equation of a line and determine the popping crease on the frame.

The system for automatically detecting No-Ball further comprises a delivery detection module. It is a binary action detection module that takes 40 consecutive frames as input and returns a binary value as a result of delivery detection. The delivery detection module works with an input model
25 that takes 40 RGB images as input with the size of 224x224x3x40 mm from data stream module, where nine 3D convolution layers with max pooling, batch normalization, and dropout are added, further three fully connected layers are added and the last layer is a fully connected layer with one neuron and sigmoid activation function as a binary classifier. This module detects when every delivery happens, so other modules can do more processing if delivery is detected, more processing
30 happens in the next module i.e. called bowler detection module.

The bowler detection module further comprises an OpenCV (open computer version) pose estimation model that is running on delivery frames to detect 17 body key points for every person in the image. Key points of a maximum number of 10 persons for 40 frames are passed to a Long Short-Term Memory (LSTM) model to detect which one is the bowler in every frame. It works with an input data model that takes 17 body key points for a maximum number of 10 persons detected in every image, for a sequence of 40 images where two LSTM layers are added. The output model of the bowler detection module is the index of the bowler for every frame if that frame contains the bowler. In a single frame, there are multiple players visible. The bowler detection module identifies and detects the bowler as the one who is doing the bowling action by applying a deep neural network model. Bowling action consists of a sequence of body postures described in figure 3. If the bowler is not detected, the model will search in the next frames to find the bowler otherwise next module is called.

The system for automatically detecting No-Ball is further comprised of a front foot landing detection module that detects the frame number that the bowler's front foot, heel lands on the ground. This module also works with an input data model that takes body key points of the bowler for a maximum number of 40 frames where one LSTM layer is added, three fully connected layers are added and the last layer is a fully connected layer with a softmax activation function and returns the bowler's front foot landing frame number.

The front foot landing detection module is responsible for detecting the bowler's front foot, heel landing moment, it looks through every frame and if front foot landing is not detected then the next frame is investigated to find the exact front foot landing moment.

It is a module for identifying the bowler's front foot, heel at the landing moment. YOLO (You Only Look Once) object detection model is retrained and used as a shoe detection model. Bowler's front foot heel at the landing moment should stay behind the popping crease to have a legal delivery. It calculates the real-world distance between the front foot heel and popping crease in millimeters. This module converts distance from pixels to millimeters using homographic transformations. If the bowler's front foot heel lands after the popping crease, the distance between the bowler's front foot heel and the popping crease has a negative value. This negative value is a flag for no-Ball. If a no-ball is reported by the previous module, the No-Ball call system is called. This module is explained in figure 6.

The system for automatically detecting No-Ball has another module called a No-Ball call system. This standard module reads the live stream from any kind of source and provides a sequence of frames for the next modules. The live match data is recorded on a database by reading from an API (Application Programming Interface) provided by the venue or done by an agent onsite. This data is recorded; the most recent ball information is always available. Ball-by-Ball data for the current ball is extracted and provided as metadata to be sent as part of the no-ball notification system. Ball-by-Ball data contains but is not restricted to the number of inning/over/ball, striker/non-striker batsman full names, and bowler full name. Also, this information is used to track the match status. No-Ball detection system takes the sequence of frames, analyzes the events and detects the bowling action, and checks if it is a legal delivery as described in Figure 5. If No-Ball is detected, Ball-by-Ball information is added to the notification message. A message is sent to the Umpire using regular widely used messaging applications like WhatsApp/Telegram or a messaging client installed on a pc/mobile depending on the Umpire's choice. The message contains Ball-by-Ball information, the exact frame that No-Ball happened, and a short clip containing that delivery. At the end of the session, a summary of the session containing the count of no-balls and their corresponding Ball-by-Ball information is reported.

BRIEF DESCRIPTION OF THE DRAWINGS

In order to better explain every module of the present disclosure, reference will now be made to the following accompanying figures.

In the figures:

Figure 1: illustrates the cricket pitch for introducing bowling action, where the bowler throws the ball to the batsman.

Figure 2: depicts the pitch diagram for defining popping crease in detail.

Figure 3: illustrates the bowling action steps, especially the front foot contact step where the bowler's front foot lands on the ground.

Figure 4: illustrates the No-Ball situation, with regard to the positioning of the bowler's feet. If no part of the bowler's front foot at front foot contact moment, whether grounded or raised is within the back edge of the popping crease.

Figure 5: illustrates the No-Ball detection flowchart, and all steps taken to detect No-Ball.

Figure 6: illustrates the No-Ball notification and tagging system on broadcast videos or live matches, with real-time consideration of ball-by-ball data.

Figure 7: illustrates the popping crease detection module or line detection module.

5 Figure 8: illustrates the delivery detection model. This module is a binary classification model that takes a list of frames as input and returns 1 if it is a delivery otherwise returns 0.

Figure 9: illustrates the bowler detection module. A recurrent neural network that takes key points of every person for a sequence of frames as input and returns the id of the bowler for every frame.
10

Figure 10: illustrates the front foot landing frame number detection module. This is a recurrent neural network model that takes bowler key points for every frame as input and returns the frame number that the front foot lands on the ground.

Figure 1 shows that the cricket pitch consists of the central strip of the cricket field between the wickets. It is 22 yd (20.12 m) long (1 chain) and 10 ft (3.05 m) wide. One player called a batsman stands at one end of the pitch and another player from the opponent team called a bowler throws a cricket ball toward the batsman.
15

Figure 2 illustrates the popping crease, which is the back edge of the popping crease marking, shall be in front of and parallel to the bowling crease and is 4 ft/1.22 m from it. The popping crease is marked to a minimum of 6 ft/1.83 m on either side of the imaginary line joining the centers of the two middle stumps and is considered to be unlimited in length.
20

Figure 3 illustrates the bowling action steps, especially the front foot contact step where the bowler's front foot lands on the ground. In the sport of cricket, the bowling action is the set of movements that result in the bowler releasing the ball in the direction of the batsman. As illustrated in the figure there are usually 7 steps for doing the bowling action by the bowler. One step is front foot contact where the bowler's front foot touches/lands on the ground.
25

Figure 4 illustrates No-Ball with regards to the positioning of the bowler's front foot. In bowling action, at the front foot contact moment as described in figure 3, while the bowler's front foot is

touching the ground, if no part of the front foot whether grounded or raised is within the back edge of the popping crease, it is a No-Ball.

Figure 5: illustrates the No-Ball detection flowchart, and all steps taken to detect No-Ball. The sequence of frames received from a live broadcast data stream module (101) or a recorded video file. They could be in any number of frame rates and any resolution, compressed in any form of compression, or provided as raw data. This module converts the stream of data to a sequence of discrete frames and feeds them to the next module line detection module (102). The main purpose of a line detection module (102) is to detect popping-crease on the cricket pitch as mentioned in figure 2. Some portion of the bowler's front foot at the landing moment should stay behind this line unless it is a No-Ball as described in figure 4. A delivery detection module (103) detects every delivery when it happens, so other modules can do more processing and if delivery is detected, more processing happens in the bowler detection module (105) otherwise next frame of delivery gets processed. In every frame, there are multiple players visible, and the bowler is detected as the one who is doing the bowling action by applying a deep neural network model. Bowling action consists of a sequence of body postures described in figure 3. If the bowler is not detected, model 106 will search in the next frames to find the bowler otherwise next module named front foot landing detection module (107) is called. This module is responsible for detecting the bowler's front foot landing moment, it looks through every frame and if front foot landing is not detected then the next frame is investigated to find the exact front foot landing moment. The heel of the foot landing identifier module (108) is a module for identifying the bowler's front foot heel at the landing moment. YOLO object detection model is retrained and used as a shoe detection model. Bowler's front foot heel at the landing moment should stay behind the popping crease to have a legal delivery. The determiner module (109) calculates the real-world distance between the front foot heel and popping crease in millimeters. This module converts distance from pixels to millimeters using homographic transformations. If the bowler's front foot heel lands after the popping crease, the distance between the bowler's front foot heel and the popping crease has a negative value. This negative value is a flag for No-Ball by no ball detected module (110). If a no-ball is reported by the previous module (110), the No-Ball call system (111) is called.

Figure 6: illustrates the No-Ball notification and tagging system on broadcast videos or live matches, with real-time consideration of ball-by-ball data. This standard module reads the live

stream from any kind of source (601) and provides a sequence of frames for the next module's live match data (602). The live match data is recorded to a database by reading from an API provided by the venue or done by an agent onsite. This data recorded the most recent ball information is always available. Ball-by-Ball data for the current ball is extracted from the match tracker module (603) and provided as metadata to be sent as part of the No-Ball notification system. Ball-by-Ball data contains but is not restricted to the number of inning/over/ball, striker/non-striker batsman full names, and bowler full name. Also, this information is used to track the match status. The event analyzer of the No-Ball detection system takes the sequence of frames, analyzes the events and detects the bowling action, and checks if it is a legal delivery as described in Figure 5.

10 If No-Ball is detected, Ball-by-Ball information is added to notification message module 605. The message is sent to the Umpire using regular widely used messaging applications like WhatsApp/Telegram or a messaging client installed on a pc/mobile depending on the Umpire's choice. The message contains Ball-by-Ball information, the exact frame that No-Ball happens, and a short clip containing that delivery.

15 At the end of the session, a summary of the session containing the count of No-Balls and their corresponding Ball-by-Ball information is reported by end of the session module (606).

Figure 7: illustrates the module for detecting the popping crease. The popping crease is a line defined in figure 2, and it can be distinguished by two points on the line. The line detection module is a landmark detection that detects two ends of the popping crease on every frame.

20 The model (701) takes one RGB image with an input size of 224x224x3mm. A convolution layer (702) is added, then 17 bottleneck residual blocks (703) are added to the model. The model (704) illustrates the structure of every residual block, that is constructed from a convolution layer, Dwise convolution, and a residual connection. The dropout layer (705) is added to have a regularization effect and two separable convolution layers (706) are added, the first layer has a
25 kernel size of 5 and an activation function of relu and the second layer has a kernel size of 3 and activation function of sigmoid, so the output of this layer has four values which correspond to coordinates of two points. The output of the model is two ends of the popping crease. This makes it possible to build the equation of a line and determine the popping crease on the frame (707).

Figure 8: illustrates the delivery detection model as a binary action detection module that takes 40 consecutive frames as input and returns a binary value as a result of delivery detection.

The model (801) takes 40 RGB (Red, Green, Blue primary colours) images (as input with a shape of 224x224x3x40mm. Nine 3D convolution layers (802) with max pooling, batch normalization, and dropout are added. Three fully connected layers (803) are added. The last layer (804) is a fully connected layer with one neuron and a sigmoid activation function as a binary classifier.

Figure 9: illustrates the bowler detection module. A recurrent neural network that takes key points of every person for a sequence of frames as input and returns the id of the bowler for every frame. The OpenCV (Open Computer Vision) pose estimation model (901) is running on delivery frames to detect 17 body key points for every person in the image. Key points of a maximum number of 10 persons for 40 frames are passed to a Long Short-Term Memory (LSTM) model to detect which one is the bowler in every frame.

The input data for the model is 17 body key points for a maximum number of 10 persons detected in every image, for a sequence of 40 images. Two LSTM layers (902) are added. The output of the model is the index of the bowler (903) for every frame if that frame contains the bowler.

Figure 10: illustrate the model that detects the frame number that the bowler's front foot lands on the ground.

The body key points of the bowler for a maximum number of 40 frames are passed to the model (1001) as input data. One LSTM layer (1002) is added. Three fully connected layers 1003 are added. The last layer (1004) is a fully connected layer with a softmax activation function and returns the bowler's front foot landing frame number.

Claims

1. A system for automatically detecting a front foot No-Ball through machine learning techniques comprising:

a data stream module (101) for providing raw data or sequence of frames from a live broadcast data or a recorded video file;

a line detection module (102) for detecting a popping crease comprising: an input model (701) that takes a RGB (Red, Green, Blue primary colours) image and adds a convolution layer (702), 17 bottleneck residual blocks (703) further adds a dropout layer (705) and two separable convolution layers (706), and an output model (707) to provide two ends of the popping crease and determining the popping crease on a frame;

a delivery detection module (103) for detecting every delivery of ball comprising:

a binary action detection module (801) that takes 40 consecutive frames and adds nine 3D convolution layers (802) and further adds a fully connected layer (804) with one neuron and sigmoid activation function as a binary classifier;

a Bowler detection Module (105) for detecting a bowler, when delivery of the ball is detected by the delivery detection module, by applying a deep neural network model comprising: an OpenCV (open computer version) pose estimation model (901) that detects body key points for every person in an image and passes to a Long Short-Term Memory (LSTM) model to detect the bowler in every frame, further adds two LSTM layers (902), and a bowler index model (903) that recognizes the bowler;

a front foot landing detection module (107) for detecting the bowler's front foot landing moment comprising: an input model (1001) that adds an LSTM layer (1002), three fully connected layers (1003) and a last layer (1004) that is fully connected layer with a softmax activation function and returns a bowler's front foot landing frame number, with the body key points of the bowler from the Bowler detection module as the input data in the input model;

a heel of the foot landing identifier module for identifying the bowler's front foot heel from the bowler's front foot landing frame number returned from the front foot landing detection module;

a determiner module to calculate a real-world distance between the front foot heel and the popping crease to determine if the bowler's front foot heel lands after the popping crease and is thus a No-Ball and to call a No-Ball call system;

and

5 the No-Ball call system (111) comprising: a notification message module (605) to send a message containing ball-by-ball information to the Umpire using regular widely used messaging applications when the No-Ball is determined by the determiner module, and an end of session module (606) to report a summary of a session containing a count of No-Balls and their corresponding Ball-by-Ball information.

10 2. The system as claimed in claim 1, wherein said data stream module (101) for providing a raw data or sequence of frames from a live broadcast data or a recorded video file is in any number of frame rate or resolution, or compressed in any form of compression.

15 3. The system as claimed in claim 1, wherein in the line detection module (102) the RGB image has a size of 224x224x3mm; the 17 bottleneck residual blocks (703) are constructed from a second convolution layer, a Dwise convolution and a residual connection; and the first separable convolution layer has a kernel size of 5 and an activation function of relu and the second separable convolution layer has a kernel size of 3 and
20 activation function of sigmoid.

 4. The system as claimed in claim 1, wherein each of the 40 consecutive frames is a second RGB image with a size of 224x224x3mm; and the nine 3D convolution layers are added (802) with max pooling, batch normalization, and dropout.

25 5. The system as claimed in claim 1, wherein the OpenCV (open computer version) pose estimation model (901) detects 17 body key points for every person in an image, wherein the key points of a maximum number of 10 persons for 40 frames are passed to the Long Short-Term Memory (LSTM) model to detect the bowler in the frame.

6. The system as claimed in claim 1, wherein the message further contains the exact frame that the No-Ball happens and a short clip containing the delivery, the message is sent using regular widely used messaging applications like WhatsApp/Telegram or messaging client installed on mobile/pc (personal computer) depending on the Umpire's choice.

7. A method for preparing the system as claimed in any one of claims 1 to 6.

8. A use of the system for automatically detecting No-Ball as claimed in any one of claims 1 to 6.