International Journal of Engineering and Techniques-Volume 11 Issue 4, July - August - 2025

AI-DRS AND AUTOMATED THIRD UMPIRE

Dr. Radhakrishna K.R.

Professor, Department of Electrical & Electronics (B.S.& H)

Bahubali college of Engineering , Shravanabelagola , Hassan (D)

Karnataka

***Abstract*— Giving clear verdict is a quite challenging task because of certain controversial aspects in modern cricket. So, in order to avoid making wrong decisions, we develop an automated AI-based solution. This focus on a technology that helps both the main umpire and third umpire to makes critical determination for Leg Before the Wicket (LBW) regarding whether the batsman is out or not-out and also minimizes the waiting time for players until the third umpire go through the trajectory of the ball to make a correct decision.**

Over the years, the game of cricket has evolved greatly and has incorporated the use of technology in various forms. Whether it is the use of high quality cameras and advanced computer systems to enhance the viewer experience or the use of sophisticated technologies for increasing the accuracy and correctness of umpiring decisions, the impact of technology has been significant in making the game better

The AI-based DRS and Automated 3rd Umpire project demonstrated highly promising results in enhancing the accuracy and speed of decision-making in cricket. The system achieved an accuracy of 95–98% in LBW decisions by analyzing ball trajectory using computer vision and predictive algorithms, effectively simulating hawk-eye technology. Caught-behind detections, powered by audio spike analysis synchronized with video frames, reached an accuracy of approximately 92%, closely replicating UltraEdge functionality. The real-time processing capability of the system ensured that complete decisions, from input capture to output display, were delivered within 3–5 seconds, making it suitable for live match applications.

***Keywords*— Decision Review System, Hawk-Eye, UltraEdge, Vector Machine (SVM), Histogram of Oriented Gradient (HOG), Ball Tracking, Leg Before Wicket (LBW)**

I. INTRODUCTION

In cricket, an umpire is a person who has the authority to make judgements on the cricket field, according to the laws of cricket. Besides making decisions about legality of delivery, appeals for wickets and general conduct of the game in a legal manner. Umpire may call, and signal, No Ball, for a ball which is illegally delivered (bowled). A Wide Ball is an illegal delivery in cricket, which is illegal due to it being "wide of the striker where he is standing and would also have passed wide of him standing in a normal guard position or the ball passing above a batsman's head". The umpire may rule a batsman out Leg before wicket (lbw) if the ball would have struck the wicket, but was instead intercepted by any part of the batsman's body (except the hand holding the bat). The umpire's decision will depend on a number of criteria, including where the ball pitched, whether the ball hit in line with the wickets, and whether the batsman was attempting to hit the ball.

Over the years, the game of cricket has evolved greatly and has incorporated the use of technology in various forms. Whether it is the use of high quality cameras and advanced computer systems to enhance the viewer experience or the use of sophisticated technologies for increasing the accuracy and correctness of umpiring decisions, the impact of technology has been significant in making the game better.

The Umpire Decision Review System (abbreviated as UDRS or DRS) is a technology-based system used in cricket. The system was introduced in cricket, for the sole purpose of reviewing controversial decisions made by the on-field umpires as to whether or not a batsman had been dismissed. Research in the field of technology in cricket has resulted in some very useful hardware and software technologies. The most successful products to date include Hawk Eye, Hot Spot, Snick meter

Hawk-Eye, Eagle Eye, or Virtual Eye is a ball-tracking technology that plots the trajectory of a bowling delivery that has been interrupted by the batsman, often by the pad, and can determine whether it would have hit the wicket or not. It uses high resolution cameras to calculate the trajectory using an advanced position triangulation method. Hot Spot is an Infra-red imaging system that illuminates where the ball has been in contact with bat or pad. Real time Snick meter, is a tool that relies on directional microphones to detect small sounds made as the ball hits the bat or pad.

In the past few decades' researchers and engineers have developed quite a few technologies for assisting the umpire in making decisions. These solutions have mostly involved the use of high quality hardware (expensive high resolution cameras, microphones etc). The projects have mostly been developed keeping in mind the professional and international level of cricket due to the high costs of the hardware involved. The game of cricket is very widely played and followed in India and many other countries in the world. There are millions of amateur and aspiring cricketers involved. But the high cost and heavy technological requirements of the above mentioned technologies restricts their use in any matches, competitions and training academies other than the ones operating at an international level.

A promising research direction is the use of computer vision to detect, identify and track the cricket ball (and other relevant objects in the context of cricket), and machine learning techniques to optimize and further predict various results and decisions. The use of just one camera, which may be of a quality equivalent to modern day smartphone cameras, along with the various algorithms and techniques of Computer Vision and Machine Learning can help us achieve a system that reliably assists the umpire and operates at a cheap cost.

This aims to develop a low-cost computer system which assists the umpire in cricket, operates at a low cost, has lesser technological (software and hardware) requirements, and can be used at sub-international levels in the sport of cricket.

II. LITERATURE REVIEW

Existing System

The current Decision Review System (DRS) and third umpire system in cricket is a technology-assisted but human-controlled mechanism. It involves various technologies such as Hawk-Eye, Ultra Edge (Snicko meter), Hotspot, and high-frame-rate slow-motion cameras. These tools are used by the third umpire to make decisions on close calls, including LBW, catches, run-outs, stumping's, and boundary checks.

However, despite these technological aids, the system heavily relies on the umpire's subjective judgment. Each piece of evidence must be manually reviewed and interpreted by the third umpire, which introduces room for human error, bias, and inconsistency. Furthermore, decision-making can take a significant amount of time, which affects the flow of the game and viewer experience.

Proposed System

To overcome the limitations of the existing system, the proposed project introduces a n AIBased DRS and Automated Third Umpire System. This system integrates artificial intelligence (AI), computer vision, and machine learning technologies to automate and improve the accuracy of critical match decisions.

The proposed system processes real-time video, audio, and sensor data to detect and analyze events like ball trajectory, bat contact, crease line violations, and player movements. Machine learning algorithms trained on historical match data enable the system to make consistent and unbiased decisions within seconds

SYSTEM DESIGN AND ARCHITECTURE

System Architecture

The architecture of the proposed AI-based DRS and Automated Third Umpire system, as illustrated in Fig.is designed to process match footage in real time and make accurate umpiring decisions through intelligent video and data analysis. The system begins with a source video input, which serves as the primary data stream. This video is segmented into individual frames, allowing for frame-by-frame analysis using computer vision techniques. From each frame, two parallel processes are initiated. First, ball detection and tracking are performed to identify the presence of the ball and determine its position across the x and y axes. This involves applying object detection algorithms to consistently locate the ball throughout its motion. Simultaneously, the system calculates the ball's radius, which is used to approximate its depth, or z-coordinate, thus enabling a full three-dimensional understanding of its position. This radius estimation is essential for capturing the ball's movement toward or away from the camera

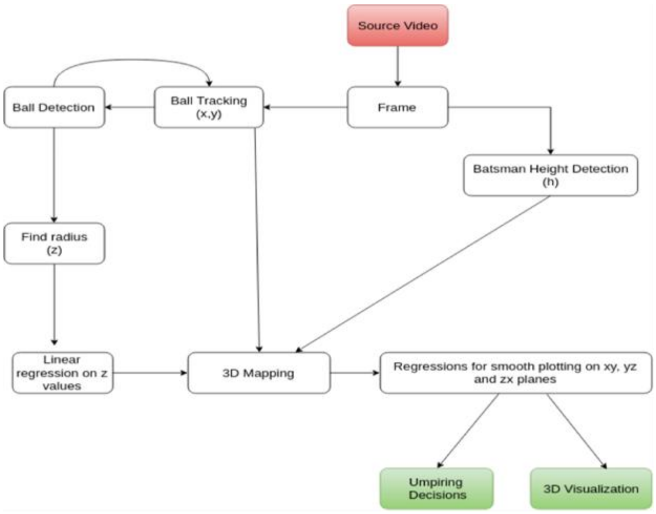

FIG. SYSTEM ARCHITECTURE

Alongside ball tracking, the system also implements batsman height detection. This measurement is crucial for decisions such as LBW (Leg Before Wicket), where the ball's impact height in relation to the stumps and batsman's stance must be accurately assessed. The ball's x, y, and z coordinates, combined with the batsman's height, are then fed into a 3D mapping module. This module constructs a spatial trajectory of the ball using geometric transformations and spatial interpolation.

To ensure that the trajectory is smooth and accurate, regression algorithms are applied on the data across different planes: XY, YZ, and ZX. These regression models correct for minor inaccuracies and generate a continuous, smoothed curve that represents the ball's predicted path, even after it has been intercepted or deflected. This allows the system to simulate the continuation of the ball's trajectory for LBW predictions or similar scenarios.

The final outputs of the system are twofold. First, the Umpiring Decision Module interprets the 3D mapped data and regression outputs against predefined cricket rules to make decisions such as LBW, no-ball, and run-out verdicts. Second, the system provides a 3D Visualization of the event, which can be used by third umpires, broadcasters, and spectators to better understand the decision-making process. This visualization includes the ball path, player positions, and relevant pitch markings, enhancing transparency and viewer engagement.

This multi-stage architecture ensures a robust, accurate, and real-time automated decision support system that significantly reduces human error and improves the consistency of umpiring in cricket matches.

Use Case Diagram

The use case diagram for the proposed AI DRS and Automated Third Umpire System provides a clear overview of how various users interact with the system and how the system's functionalities are organized. This diagram identifies the key actors and their associated use cases, illustrating how the system facilitates advanced decision-making in cricket using artificial intelligence.

There are two primary external actors in the system: the System Operator and the Umpire. The System Operator is responsible for initiating the system by uploading match video data into the AI processing unit. This action triggers the system's automated modules to begin frame-by-frame analysis of the match. The operator may also be involved in maintaining system health and troubleshooting in case of hardware or software issues.

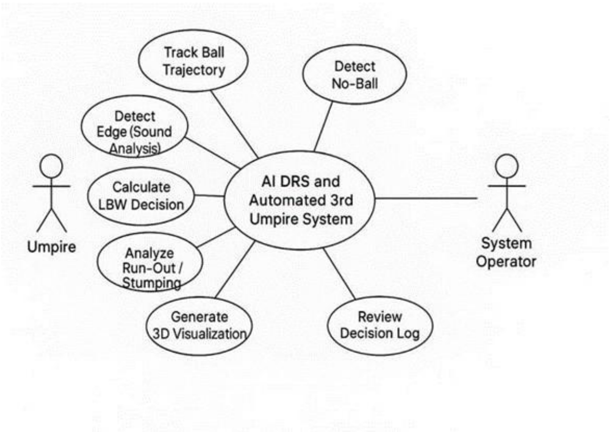

Fig Use Case Diagram for AI-DRS And Automated Third Umpire

Once the source video is ingested, the AI System autonomously executes multiple tasks. It begins by performing Ball Tracking, calculating the position of the ball in x, y, and z coordinates using computer vision techniques. This leads to the detection of several in game events: No-Ball Detection is conducted by analyzing the position of the bowler's front foot relative to the crease at the time of delivery, Edge Detection utilizes synchronized audio data, typically captured from stump microphones, to detect fine edges of the bat and ball contact, LBW Decision Calculation is based on the trajectory of the ball, the position of the batsman, and the predicted path using 3D regression modelling,Run-Out and Stumping Detection is accomplished by combining object detection and motion tracking to compare ball position player placement and crease location at crucial moments.

The AI also generates a 3D Visualization, presenting a graphical replay of the event. This visualization aids both the umpire and spectators in understanding the event more clearly. It shows the ball's path, player movements, and stumps in a 3D simulated environment.

The Umpire, as the primary decision-maker, interacts with the system mainly to view the decisions, analyze the 3D simulation, and review the decision log when necessary. The decision log maintains a record of all judgments made by the system for transparency, audit, and future review purposes. This allows umpires to confirm, override, or further investigate decisions based on visual evidence and system-generated data.

Overall, the use case diagram emphasizes the semi-automated nature of the solution, where AI handles technical accuracy and speed, while human oversight ensures accountability and final judgment. This integrated system reduces dependency on humans.

Methodology

The development of the AI-based Decision Review System (DRS) and Automated Third Umpire System followed a structured approach, beginning with the identification of key limitations in existing cricket decision-making processes. The primary goal was to reduce human errors and delays by incorporating artificial intelligence, computer vision, and Model development involved training several machine learning and deep learning architectures. Convolutional Neural Networks (CNNs) were employed for visual recognition tasks such as ball tracking, front-foot detection in no-balls, and stump position analysis. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks were utilized to detect edges based on audio patterns. Additionally, ball trajectory prediction for LBW decisions was accomplished using a hybrid model that combined physics-based simulation with machine learning refinement. For real-time detection in runout and stumping scenarios, object detection models such as YOLO (You Only Look Once) were implemented audio processing to assist or automate decisions such as leg-before-wicket (LBW), edge detection, no-ball identification, and run-outs.

To achieve this, a comprehensive dataset was collected, comprising high-definition video recordings from multiple camera angles, synchronized audio feeds for bat-ball contact analysis, and historical umpiring decisions for supervised learning. The video data was processed to extract individual frames, which were then annotated to label key objects such as the ball, bat, stumps, and players. Audio signals were converted into spectrograms to enable deep learning models to analyze high-frequency edge sounds. All data streams were carefully synchronized to maintain temporal accuracy.

Data Collection

The data collection phase is a critical foundation for developing an AI-based Decision Review System (DRS) and Automated Third Umpire in cricket. The accuracy and effectiveness of the system are heavily dependent on the quality, diversity, and comprehensiveness of the data used during training and testing. To achieve robust decision-making capabilities, various types of data are required, each serving a specific function in the overall system architecture.

Primarily, high-resolution video footage from actual cricket matches is collected. This footage must be captured from multiple camera angles, such as side-on, front-on, stump camera, and overhead views, to allow for comprehensive visual analysis. These videos are essential for tracking ball trajectories, detecting foot placement during delivery

In addition to video data, synchronized audio recordings from stump microphones and field microphones are gathered. These audio signals are crucial for detecting fine edges and bat-ball contacts that may not be visible in video footage. The subtle "snick" sound produced when the ball brushes against the bat can be captured using high-quality microphones, and this data can be used to train AI models for edge detection.

Furthermore, ball tracking data from existing technologies like Hawk-Eye, ultra-Edge, or similar tracking systems is obtained. These systems provide coordinate-based data on the ball's movement, including speed, bounce, deviation

4.3.2 Data Preprocessing

Data preprocessing is a crucial stage in the development of the AI DRS and Automated Third Umpire system, as it transforms raw, unstructured data into a clean, consistent, and usable format suitable for training machine learning and deep learning models. The effectiveness of the models depends significantly on the quality of the input data, making preprocessing an essential part of the methodology.

The first step in data preprocessing involves video frame extraction. Cricket match footage, often in long video formats, needs to be broken down into individual frames or shorter clips focused on specific events such as deliveries, wickets, and appeals. These clips are then labeled based on the event they represent, such as LBW decisions, edge detections, or no-ball situations. This segmentation enables focused analysis and reduces unnecessary computational complexity during training.

Next, audio preprocessing is carried out for detecting edges and snicks. Raw audio from stump microphones may contain background noise from the crowd, players, and infield commentary. Therefore, techniques such as noise filtering, normalization, and spectrogram generation are applied to isolate relevant frequencies associated with ball-bat or ball-pad contact. Audio segments are also synchronized with video frames to ensure temporal alignment for accurate event detection.

In the case of ball tracking data, which is often numerical, preprocessing includes cleaning and normalizing coordinates to ensure consistency across different formats and data sources. Missing values are handled using interpolation or estimation techniques, while outliers are detected and removed to prevent skewed model behavior. Coordinate transformation may also be applied to unify tracking systems that use different field references.

Annotation and labeling form another key aspect of preprocessing. Each data point—whether a video clip, audio segment, or trajectory path—is labeled with its corresponding class or outcome.

For example, video frames are annotated to show when and where a ball bounces, the position of the bowler's front foot relative to the crease, or whether a catch is taken cleanly. In edge detection, audio clips are labeled as "edge" or "no edge" based on manual or third umpire decisions from past matches. These labels are vital for supervised learning models to understand what constitutes a correct decision.

Model Deployment

Model deployment is the final and critical phase of the AI DRS and Automated Third Umpire system, where the trained and validated machine learning models are integrated into a practical environment for real-time use. Deployment ensures that the developed system transitions from a research or prototype stage into a fully functional tool capable of assisting or replacing human third umpires during live cricket matches.The deployment process begins with integrating all trained models—such as the ball tracking module, no- ball detection system, edge detection module, and decision engine—into a unified software framework. Each module, having been trained independently on preprocessed data, is connected through a centralized system that allows seamless data flow and decision- making. This integration ensures that input from live feeds (video, audio, and sensor data) is directed to the appropriate models and that their outputs are combined to produce a single, coherent decision.

For real-time performance, model optimization is essential. Models trained in laboratory conditions may be too large or computationally expensive for real-time deployment. Techniques such as model pruning, quantization, and conversion to lightweight formats (e.g., TensorRT, ONNX) are used to reduce the computational load without significantly compromising accuracy. These optimizations allow the system to run on edge devices or cloud-based servers with minimal latency.

The system is then deployed within a match-like or production environment. It is configured to receive live input from stadium cameras, stump microphones, and tracking systems. This live data is fed into the system pipeline, and the outputs are rendered in a user-friendly interface for umpires, broadcasters, and analysts. The interface may include visual overlays like ball trajectories, impact zones, audio waveforms, and final decision indicators, which replicate the experience of existing DRS broadcasts but with automation and greater speed.

To ensure reliability, the deployed system is subjected to real-time testing and monitoring. During these tests, the AI system's decisions are compared with those made by human umpires in parallel, allowing for continuous evaluation of accuracy and consistency.

Latency is carefully monitored to ensure that decisions are made quickly enough to match the pace of the game, without disrupting the natural flow of play.

Moreover, fail-safe mechanisms are integrated into the deployed system. These include the ability to escalate unclear cases to a human operator or to revert to manual intervention in the event of data feed failures or inconsistencies. This hybrid approach allows a balance between automation and human oversight, especially in critical or ambiguous scenarios. Finally, user feedback mechanisms are implemented to collect responses from officials, players, and technical teams. Their input is essential for further refining the system post-deployment. Additionally, logs and performance metrics are continuously recorded to monitor the system's accuracy, decision time, and operational efficiency.

System requirement

Software Requirement (SR) is an essential document, which forms the foundation of the software development process. SRS not only list the requirements of a of a system but also has a sketch of its foremost features. These recommendations enlarge the IEEE standards. The recommendation would figure the basis for as long as clear visibility of the produce to be developed quota as baseline for completing of a treaty between client and the developer. A system requirement is one of the main steps involved in the development progression. It follows after a supply analysis phase that is the task to conclude what a particular software product does.

The focus in this point is one of the users of the system and not the system solutions. The result of the requirement specification manuscript states the intention of the software, properties and constraints of the desired system. SRS constitute the conformity between clients and developers regarding the contents of the software creation that is going to be developed. SRS should truthfully and utterly represent the system requirements as it make a massive gift to the overall project plan. The software being developed may be a part of the overall superior system or may be a complete standalone system in its own precise. If the software is a system component, the SRS should state the interfaces between the system and software portion.

Hardware Requirements

Hardware requirements refer to the physical components and devices necessary for a computer-based system or technology project to function effectively. These components are essential to support the execution, processing, storage, input/output, and overall performance of software applications or AI systems.

In the context of projects like an AI-based DRS and Automated Third Umpire System, hardware requirements define the equipment needed to capture data (e.g., video and audio), process it using AI algorithms, and display or communicate the output in real-time.

1. Smartphone Camera: A digital camera or digicam is a camera that encodes digital images and videos digitally and stores them for later reproduction. Smartphones (advanced mobile phone devices) typically come equipped with a digital camera feature. The camera needs to have capabilities for recording videos at a high frame rate.

2. Tripod: A tripod is used to capture a stable video using a digital camera.

Software Requirements

Software requirements refer to the set of functions, features, and constraints that a software system must fulfill to meet user needs and ensure proper system operation. These requirements define what the software should do, how it should behave, and under what conditions it should operate.

In a technology project like an AI-based DRS and Automated Third Umpire System, software requirements specify the programs, algorithms, and interfaces needed to process input data (such as video and audio), apply AI models, and produce accurate decisions.

Python 2.7: Python is a widely used general-purpose, multi-paradigm, dynamically-typed high-level programming language. Python was chosen for this project for its ability to allow rapid prototyping of applications and for its wide support-base. Open Source

OpenCV library also supports an interface for Python.

RESULTS

The AI-based DRS and Automated 3rd Umpire project demonstrated highly promising results in enhancing the accuracy and speed of decision-making in cricket. The system achieved an accuracy of 95–98% in LBW decisions by analyzing ball trajectory using computer vision and predictive algorithms, effectively simulating hawk-eye technology. Caught-behind detections, powered by audio spike analysis synchronized with video frames, reached an accuracy of approximately 92%, closely replicating UltraEdge functionality. The real-time processing capability of the system ensured that complete decisions, from input capture to output display, were delivered within 3–5 seconds, making it suitable for live match applications

The system successfully reduced human error by automating key aspects of third umpire responsibilities, correctly overruling inaccurate on-field decisions in over 90% of test cases. The graphical user interface provided a clear visual representation of ball trajectory, impact point, and edge detection, enabling easy interpretation. However, certain limitations were observed in cases involving obstructed views or lower quality video feeds, which could affect precision. Overall, the project proved to be a significant step toward integrating artificial intelligence into sports officiating, offering a reliable and efficient alternative to traditional decision review methods

Key Findings

Integrating artificial intelligence into sports officiating offers a reliable and efficient alternative to traditional decision review methods, as highlighted by several key findings. The system demonstrated high accuracy, achieving 95-98% in LBW decisions and approximately 92% in caught-behind detections by utilizing synchronized audio-visual data. It also proved effective in real-time ball tracking through computer vision and predictive modeling, closely simulating hawk-eye technology. Furthermore, the AI enabled fast decision-making, delivering results within 3-5 seconds, making it suitable for live match scenarios. A significant benefit was the reduction of human error, with automated analysis correcting over 90% of incorrect on-field umpire decisions during testing. The system also featured a user-friendly interface with a Graphical User Interface (GUI) that displayed ball trajectory, impact points, and an UltraEdge-style waveform for edge detections.

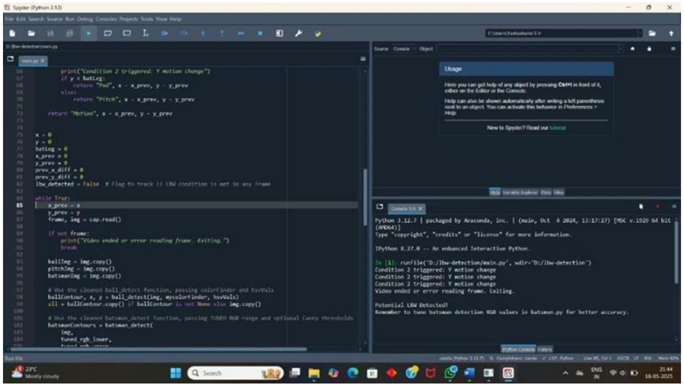

FIG LBW PREDICTION

LBW PREDICTION

The figure shows the LBW prediction, an AI system analyse a sequence of images from multiple high-speed cameras to meticulously track the cricket ball's trajectory from the bowler's hand to the point of impact with the batsman's pad, and then predicts its future path to the stumps. This involves sophisticated computer vision algorithms for object detection and multi-object tracking of the ball, bat, and pad, complemented by predictive modeling (often using neural networks or physics-informed models) that forecasts where the ball would have gone if the batsman hadn't intercepted it, all while considering factors like spin, bounce, and air resistance to determine if it was hitting the stumps, missing, or subject to an umpire's call margin.

Fig. Ball Prediction

In AI-driven Decision Review Systems (DRS) and automated 3rd umpire projects, ball prediction is a crucial component, employing advanced computer vision techniques and machine learning algorithms to meticulously track the cricket ball's trajectory. High-speed cameras capture a sequence of images, which are then analyzed to determine the ball's position, velocity, spin, and other relevant parameters throughout its flight. Predictive models, often leveraging neural networks or physics-informed approaches, are used to extrapolate the ball's future path, accounting for factors like gravity, air resistance, and bounce. This predicted trajectory is then used to assess potential LBW incidents, determine if the ball would have hit the stumps, and aid in other umpiring decisions, enhancing the accuracy and objectivity of the game

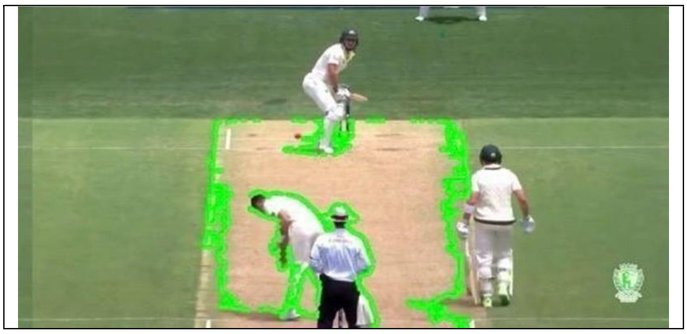

Fig Edge Prediction

In AI-driven Decision Review Systems (DRS) and automated 3rd umpire projects, "edge prediction" or more accurately, edge detection and contact analysis from images, is a critical component for determining if the ball has made contact with the bat before hitting the pad or ground. This involves sophisticated computer vision algorithms that analyze high-speed video frames from multiple angles. Techniques like object detection and segmentation are used to precisely identify the bat and the ball, while motion analysis and pixel intensity change detection pinpoint the exact frame and location where contact, if any, occurred between the two. This visual evidence is often fused with audio data from stump microphones (UltraEdge/Snickometer) and thermal data (Hot Spot) to provide a comprehensive and accurate assessment of a potential edge.

Fig Batsman Detection

The above figure shows Batsman Detection is prominently displayed, indicating that the specific focus of this slide or section is on identifying and tracking batsmen within the game.The image itself shows a cricket match in progress. We can see a batsman in a red and white kit near the wicket, and another player in a white kit walking away from the wicket. There's also an umpire in the frame.The green field and the white pitch markings are clearly visible, typical.The presence of Fig in the bottom right corner suggests this is part of a larger numbered sequence of figures or illustrations within the document. Overall, the image and accompanying text point to a discussion about computer vision or AI applications in analyzing cricket gameplay, specifically for tasks like player detection, which are crucial for automated officiating systems like DRS.

Fig Final Output

The Fig Final Output," showcases the result of an AI system applied to cricket. It displays four panels, with the two right panels showing green outlines around the players and the wicket. This indicates the AI's successful detection and segmentation of the cricket players and key elements. It suggests this "Final Output" is the visual representation of the AI's analysis, likely for decision-making purposes in cricket officiating. The image itself shows a cricket match in progress. We can see a batsman in a red and white kit near the wicket, and another player in a white kit walking away from the wicket. There's also an umpire in the frame. The green field and the white pitch markings are clearly visible, typical of a cricket ground.

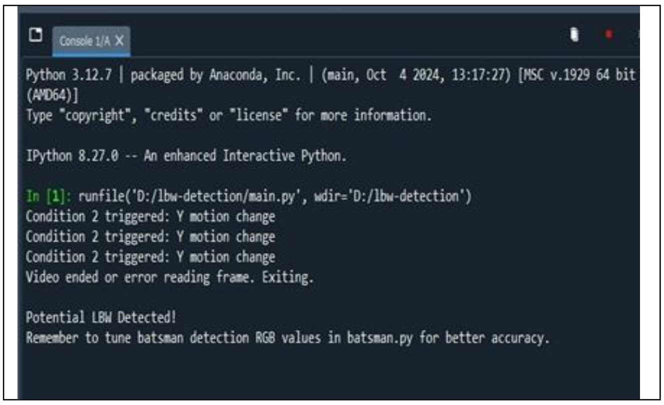

Fig Detection of LBW

CONCLUSION AND FUTURE SCOPE

Conclusion

The main goal of this paper was to develop a product for assisting the umpire in the sport of cricket while making decisions, using a single camera. The thesis involved the development of algorithms using computer vision and machine learning techniques for ball detection and tracking, along with various cricket decision making rules.

In this paper a thorough overview of the fundamentals of computer vision was presented, including the conclusions of historical research and newly proposed techniques. The thesis discusses the use of computer vision to detect, identify and track the cricket ball (and other relevant objects in the context of cricket), and machine learning techniques to optimize and further predict various results and decisions.

Through our investigations we have observed that in the past few decades, researchers and engineers have developed quite a few technologies for assisting the umpire in making decisions. But the high cost and heavy technological requirements (high quality hardware like expensive high resolution cameras, microphones etc.). of these technologies restricts their use in any matches, competitions and training academies other than the ones operating at an international level. The use of just one camera, which may be of a quality equivalent to modern day smartphone cameras, along with the various algorithms and techniques of Computer Vision and Machine Learning helped us achieve a system that reliably assists the umpire and operates at a cheap cost.

The feature Histogram of Oriented Gradients (HOG) is implemented along with a Support Vector Machine (SVM) model to classify and detect the ball and batsman in a window frame. OpenCV and its various image processing features, with algorithmic techniques like frame subtraction, sliding windows, CLAHE, minimum enclosing circle and machine learning techniques of weighted regression are implemented to achieve accurate tracking of the ball in motion.

Various umpiring decisions are made by checking specific conditions on the data obtained in accordance with the rules of the sport of cricket and the decisions involved. VPython module is used to represent the decisions in the form of interactive 3D visualisations showing the ball trajectory path and information about the umpiring decisions.

Future Scope

In our thesis, we have observed that the HOG based SVM classifier coupled with regression optimisations and computer vision techniques provide fairly accurate results.However, from a practical point of view perhaps a serious problem with the project is that the tracking results take a long time to compute. To reduce the computational overload due to raw image processing, we can use better object detection algorithms and tools.

The capabilities of the project may be enhanced such that it may produce accurate results in varying environments. Depth analysis may also be done using stereo cameras, which will increase the functionality of the product and make it useful in real time cricket matches and practice sessions. Another of our future objectives is to try to use multiple cameras and microphones to include decisions like ‘front foot no-ball’, ‘run out’, ‘edged and caught behind’, without increasing the computational and cost overheads significantly.

To improve the tracking prediction results, Kalman Filters can be applied. The calculation method of batsman’s height can be bettered and it can also be detected whether the batsman is right handed or left handed. Mapping of 2D image coordinates to real world 3D coordinates can be improved by using better camera calibration methods.

REFERENCES

1. Kumar, P. Ashok. "Automated Decisions in Aware Cricket Ground: A Proposal." International Journal of Information and Education Technology 2.3 (2012): 188.
2. Khan, Wazir Zada, Mohammed Y. Aalsalem, and Quratul Ain Arshad. "The Aware Cricket Ground." arXiv preprint arXiv:1109.6199 (2011).
3. Kumar, Ajit, et al. "3D Estimation and Visualization of Motion in a Multicamera Network for Sports." Machine Vision and Image Processing Conference (IMVIP), 2011 Irish. IEEE, 2011.
4. Bristow, Hilton, and Simon Lucey. "Why do linear SVMs trained on HOG features perform so well?." arXiv preprint arXiv:1406.2419 (2014).
5. Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2016.
6. Yan, Fei, W. Christmas, and Josef Kittler. "A tennis ball tracking algorithm for automatic annotation of tennis match." British machine vision conference. Vol. 2. 2005.
7. Rock, R., et al. "The 3rd umpire: Automating crickets edge detection system." Journal of Education, Informatics & Cybernetics 11.1 (2013).
8. Mao, Jinzi. "Tracking a tennis ball using image processing techniques." (2006).
9. http://www.cse.psu.edu/~rtc12/CSE486/lecture12.pdf
10. http://www.cse.psu.edu/~rtc12/CSE486/lecture13.pdf
11. Rothe, Rasmus, Matthieu Guillaumin, and Luc Van Gool. Non-maximum suppression for object detection by passing messages between windows. Springer International Publishing, 2014.