2023 17th International Conference on Open Source Systems and Technologies (ICOSST) | 979-8-3503-8132-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICOSST60641.2023.10414235

Multi-Modal Architecture for Cricket Highlights Generation: Using Computer Vision and Large Language Model

Husnain Sattar
School of Computing, NUCES Islamabad, Pakistan
i211354@nu.edu.pk

Muhammad Shamil Umar
School of Computing, NUCES Islamabad, Pakistan
i211786@nu.edu.pk

Eeman Ijaz
School of Computing, NUCES Islamabad, Pakistan
i211381@nu.edu.pk

Muhammad Umair Arshad
School of Computing, NUCES Islamabad, Pakistan
umair.arshad@nu.edu.pk

***Abstract*—Generating highlights for cricket matches is a labour-intensive task that necessitates a high level of both cricket and video editing knowledge. Creating a coherent video with smooth transitions involves sorting through hours of video information, identifying key moments, and merging clips. Sports video summarization has gained a lot of traction in recent days. In this study, we provide a multi-modal framework designed for efficiently producing cricket highlights. We focus on identifying key events while utilizing information from commentary text and visual data. We make use of cues like replays, bowler and umpire positions, and commentary to do so. Starting by splitting the target video into its building blocks (non-replay deliveries), the commentary is transcribed using Automated Speech Recognition (ASR). The textual commentary is then preprocessed so as not to alter the context of the extracted speech. Based on the preprocessed text, a Large Language Model (LLM) is used to predict whether an event occurred after a particular delivery. Two computer vision models—one designed for bowler detection and the other focusing on replay identification—work at the heart of this architecture. These models perform admirably, as evidenced by their respective F1 scores of 0.97 and 0.99. Using BERT LLM exceptional F1 score of 0.96 is achieved. Notably, the architecture's large-scale training data (CricPulse) includes cricket matches from both the Indian Premier League (IPL) and Pakistan Super League (PSL), demonstrating its adaptability and robustness. In short, our study addresses the challenges of highlights generation by introducing a comprehensive framework for cricket match summarization. We help to accelerate this complex task by utilizing multi-modal inputs and cutting-edge transformer-based models, thereby improving viewing experiences for cricket lovers around the globe.**

***Index Terms*—Large Language Models, Multi-modality, Automatic Speech Recognition, Video-Summarization, Transformers**

I. INTRODUCTION

Sports highlights generation involves the automated extraction and compilation of pivotal moments from sports events into concise video segments. The availability of televised matches on websites like YouTube has made it difficult for cricket fans to keep up with the numerous updates [16]. Match highlights have become an essential medium that allows fans to stay informed without wasting time. However, manually producing these highlights is time-consuming, requires professional editing skills, and places restrictions on the processing and distribution of video. The growing amount of video material has inspired researchers in the fields of computer vision [19] and deep learning[17] to improve models like YOLO (You Only Look Once) [6] and SSD [10] (Single Shot Detectors) to increase object detection[21] speed and accuracy. In this study, our work looks into the world of cricket, developing a multi-modal [2] architecture that identifies crucial moments and then uses them to generate highlights. Fig. 1 illustrates our overall framework. An automated system for summarizing lengthy cricket matches would not only improve the viewer and editorial experience but also open up new possibilities for further research. Prior studies have analyzed ball and player tracking, audio cues, and frame clustering, but each method has had its drawbacks. Our innovative method uses a multi-modal strategy to identify important cricket occurrences. We extract crucial events like deliveries, wickets, sixes, and fours by utilizing this method. We use the YOLOv8 model to detect legal deliveries and BERT[18] to classify each ball into as an "event" (six, four and out) or "non-event". This study aims to contribute in:

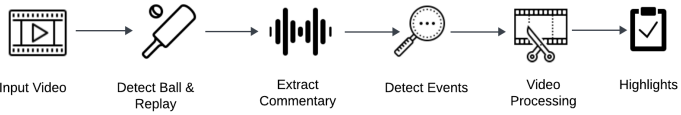

Fig. 1: Figure outlines the basic flow of architecture. We begin by detecting non-replay deliveries throughout the whole match. Next, we extract commentary related to these deliveries which is used for the detection of key events. Finally, all key events are compiled together to form a complete highlight.

- Curation of a dedicated dataset (CricPulse) tailored for YOLOv8 and BERT models. It includes the manually annotated 6000 images and the painstaking labelling of more than 22000 chunks of commentaries tied to specific ball occurrences. This dataset builds a solid base for model training, greatly increasing their accuracy.
- We propose a robust and one-of-a-kind system that utilizes multi-modal learning and speech transcripts to

produce cricket highlights. This skillful condensing of extensive gameplay keeps its depth of knowledge.
- Advocating the use for LLMs, we classify our deliveries into events using BERT. Thus, effectively using context to tackle the subjectivity present in cricket matches.

II. RELATED WORK

Generating highlights for cricket matches is an application of video sumamrization. [22]. We have seen a considerable amount of work done on basketball [1, 7], football[3, 8] matches. However, cricket, a complicated and multifaceted sport, has been the topic of extensive research in recent years [14, 16].

The event-driven and excitement-driven paradigms are the two dominant approaches in cricket highlight generation. In the former, the emphasis is on identifying crucial events, such as batsmen's strokes [5] that result in boundary hits, catches or ball tracking to locate frames that contain boundary occurrences [15]. Some approaches even use dynamic scorecard analysis [12, 16], which takes advantage of changes in the scorecard to pinpoint crucial periods. It's noteworthy that a prior work provides a technique to combine Optical Character Recognition (OCR) for textual score extraction with image averaging to track scorecard changes.

In contrast, the excitement-driven strategy relies on tracking energy levels [2] to extract important occurrences. Evidently, when players cross the line or someone takes the wicket, the audience reacts by changing its enthusiasm. Investigating these strategies is challenging. Energy levels differ significantly depending on whether a team is playing at home or away during a match. When the ball is in the air, further difficulties with ball tracking arise since the ball's color tends to blend in with the surrounding colors, making computer vision models difficult to use [11].

After a thorough research we deduced that transcribing videos for key-event extraction is progressively inclining towards popularity. [12] discusses objective evaluation methods for video summaries. This study categorizes sports content into event-based programs where a sequence of events and non-events occur. Different evaluation methods discussed in this paper include time scale modification, frame clustering by dimensionality reduction, multiple information streams and speech transcription. Our study is based on the ideas of multiple information streams (multi-modal features) and speech transcription.

Due to these subtleties, our method is in line with the breakdown of matches into their basic components using visual cues. However, our focus is on textual elements, particularly comments, as event identifiers. By utilizing the textual layer's potential to interpret crucial match junctures, this subtle strategy navigates the obstacles brought on by energy level fluctuation and ball-tracking issues.

In essence, this study emphasizes the complexity of cricket highlight generation and the variety of techniques used to condense matches into easily palatable formats. We aim to contribute to a deeper knowledge of efficient highlight extraction from cricket matches by navigating the complexities of energy dynamics, ball tracking, and commentary analysis.

Corpus Statistics of Cricket commentary	
Total Chunks	22268
Total Words	1005258
Maximum Length of Chunk	212
Minimum Length of Chunk	18
Average Length of Sentences	45
Chunks labelled as "Four"	6939
Chunks labelled as "Six"	3164
Chunks labelled as "Out"	3201
Chunks labelled as "Others"	8964
Total Events	13304

TABLE I: Statistical analysis of textual dataset

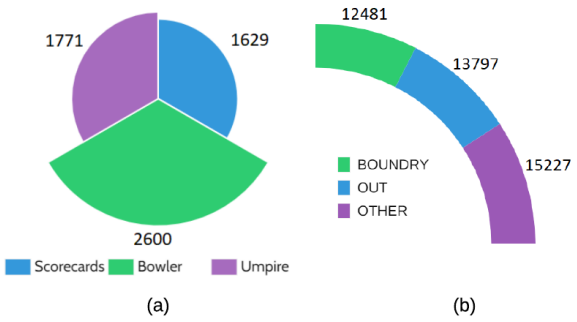

Fig. 2: Two pie charts for label distributions of the training dataset: (a) shows the distribution of labels for the YOLO model. Bowlers make up 2600 labels while the umpire and scorecards are labeled 1771 and 1629 times, respectively. (b) shows the occurrence of event-defining keywords in the dataset for the BERT model. Here, 12481 words describe a boundary while out and other events are described by 13797 and 15227 words.

III. DATASET CREATION

To comply with the requirements of the YOLOv8 and BERT Model, a diverse dataset, named CricPulse, was created. Using this large-scale data as the primary training resource, the YOLOv8 model and the BERT Model were both trained and optimized. For the YOLOv8 model, a total of about 50 matches from the Pakistan Super League (PSL) and the Indian Premier League (IPL) were put together.

A split method, specifically designed for the YOLOv8 model, was taken which is illustrated in Fig. **??**. One model was created specifically for ball detection, and the other for replay detection. A total of 2600 images were labelled as Bowler and 1771 images were labelled as umpire (refer to Fig. 2). These photographs captured the bowler and umpire near the pitch during the bowler's run-up phase. On the gathered images, a skilful filtration method was used to enhance the

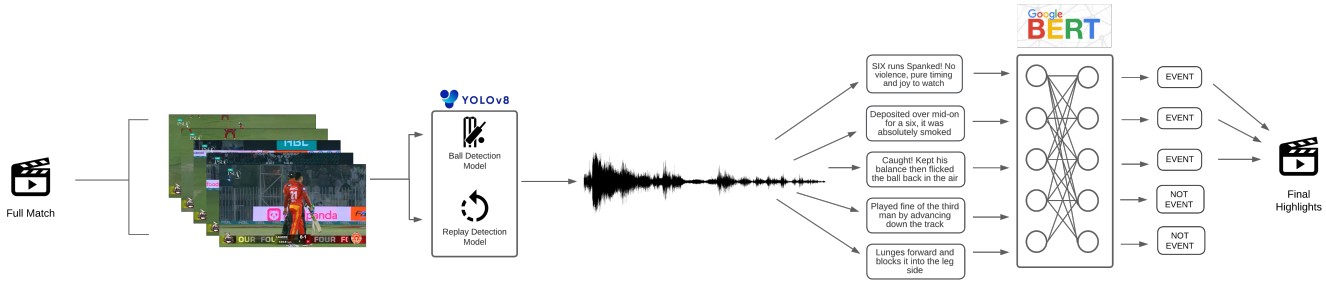

Fig. 3: Detailed diagram representing the proposed pipeline. Full match video is divided into multiple frames and each frame is passed to the YOLOv8 model to detect if delivery is valid or not. After verification, the commentary is extracted and sent to fine-tune the BERT model for classification. After classification, labels that fall under the event category are clipped and merged to form a complete highlight.

model's accuracy. We had to make sure that every image gathered was not blurry and, that both the umpire and bowler were visible along with the scorecard at the bottom. We also included mirrored frames that featured bowlers to ensure a balanced representation on both sides of the wicket.

Image annotations were completed with the help of Roboflow[4], a cutting-edge annotation tool, to support the model's bowler detection abilities. A polygonal annotation technique was used because of the bowler's natural movement. On the other hand, a rectangular bounding box worked well for the umpire's stationary presence.

A second YOLOv8 model was used for replay detection. In this regard, another well-collected dataset of 1629 images that prominently displayed scorecards placed at the lowest portion of the screen was assembled. The scorecards for different cricket leagues have varying dimensions. A rectangular bounding box method allowed us to neatly carve out the scorecards.

Proceeding the above steps we began the extraction of commentaries relative to the starting times for each delivery. The Whisper-base model was used to extract commentary from predetermined 20-second time intervals that began with the observed ball initiation incidences. This produced a corpus of over 12,600 distinct commentary snippets. To guarantee the accuracy of literary representations, each of these fragments underwent rigorous transcription. Meticulous manual

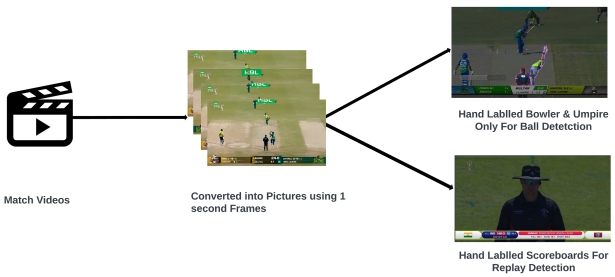

Fig. 4: Labelling process of data for YOLOv8 models. Screenshots are obtained at an interval of one second from the target match. These images are then filtered and labelled.

labelling was started at the same time. Events that corresponded with key occurrences in the retrieved commentary chunks—specifically, sixes, fours, and wickets—were systematically labelled as "EVENT." On the other hand, commentary portions lacking these crucial moments were labelled as "OTHER." This labelling technique gave the dataset contextual significance, increasing its usefulness for later analysis.

Through skilful web scraping techniques, an extra 10,268 rows of commentary were collected to broaden the dataset's coverage and inclusivity. For the creation of this additional dataset, a variety of websites containing commentary on cricket matches were used. The resulting thorough amalgamation ensured that the dataset accurately reflected a wide range of cricket matches, scenarios, and commentary.

IV. METHODOLOGY

A. *Ball-by-Ball Breakdown*

The proposed strategy for computing highlights is illustrated comprehensively in Fig. 3. To build up a highlight video we first need to deconstruct the whole match into its building blocks (deliveries or balls). To do this, we start by dividing the full match footage into a series of frames using a 0.5-second time interval. This allowed us to avoid unnecessary breaks, inning transitions, and sponsorship segments, improving both efficiency and memory management.

Considering intervals of 1 second, we noticed a comparatively high chance that crucial run-up frames will be missed. Whereas, intervals of 0.25 seconds or less produced more precision at the expense of processing time. After carefully testing and validating, we concluded that a 0.5-second interval strikes the perfect balance, ensuring an appropriate number of frames for processing and also maintaining accuracy.

After dividing the match into frames of possible deliveries, the YOLOv8 model uses the position and pose of the umpire and bowler to detect whether it is a keyframe (delivery frame) or not. This phase of the project necessitated the thorough development and evaluation of two object-detection models. In our initial model, we only attempted to predict the existence of the bowler from the run-up position. This method produced many false positives, classifying any component that even

(a) (b) (c) (d) (e)

Fig. 5: An example of how a valid delivery is detected. Once the bowler and umpire are located in frame (b), the remaining frames are used to verify if the detected one is a non-replay delivery or not.

somewhat resembled the bowler's shade as a run-up instance. Then we trained another model that would simultaneously identify the bowler and the umpire inside a single frame and because it was trying to look for two objects in each frame instead of one, it turned out to be very accurate. Moreover, different shades of colour worn by both the umpire and bowler further facilitated the model.

B. Replay Detection

After run-up frames have been successfully identified, our next goal is to classify that frame as either replay frames or non-replay frames [14]. This distinction is of utmost importance because it helps to reduce repetition in our final highlights. Fig. 4 shows one way to identify replay situations. We can quickly categorise a specific ball as non-replayable by spotting scorecards. Conversely, it would be considered replay if scorecards weren't found. Notably, scorecards appear in a variety of sizes, positions, and designs. Our initial efforts included the identification of several scorecard categories, however, these produced a sizable number of false positives.

Our model incorrectly identified batsmen's summaries as scorecards in case of dismissals. Furthermore, in replay circumstances, the sight of a small card with the word "REPLAY" on it also caused our model to trigger misclassification. Upon further research, we decided to take into account scorecards that are displayed at the bottom of the screen. This is because in modern matches the scorecard is displayed at the bottom of the screen, providing enough room for the presentation of over and player run counts, bowling statistics, targets and much more. Additionally, this strategic focus greatly improved the precision of our overall pipeline. Let B be the frame at which the bowler is detected and R be the times scorecard is detected in the next 6 frames:

$$Delivery = \begin{cases} Valid & : \ B \cap (R \leq 3) \\ Invalid & : \ B \cap (R \geq 3) \end{cases}$$

If in the next six frames, the scorecard is detected more than 3 times, that delivery will be labeled as a non-replay ball otherwise it will be labeled as a replay. Fig. 5 gives an example of how a valid time frame for each ball is found.

C. Commentary Extraction and Key-Event Detection

For extracting commentary text from match footage, we needed an audio transcriber[9]. Whisper, an advanced automatic speech recognition (ASR) model, is one of the fastest and most accurate audio transcribers available today. It was developed and open-sourced by OpenAI, trained for more than 100,000 hours of audio data obtained from the web. Whisper has multiple versions out of which we used the help of whisper-large[13] for our architecture. Its impressive capabilities were crucial in accurately converting cricket commentary from audio samples into a written copy.

In our method, the audio was divided into several chunks, each of which represented a valid delivery. Each piece was 20 seconds long and covered the commentary from the beginning of the delivery to the next 20 seconds. The whisper-large model's ability to transform was used to skillfully translate the audio into text. The conversion to lower case of all words in the obtained text was the fist step to cleaning of the data. We perform minimal pre-processing steps to preseve the commentary's orginal temporal instances which could enhance the performance of our proposed framework. However, there a few instances that had to be removed because the commentary is deviated from English. This limitation resulted from the fact that we only use the BERT model that has been fine-tuned on English commentary data[20]. Then, we ran each commentary chunk through our adjusted BERT model. This model's proficiency in language classification was proved after it underwent thorough adaptation using a dataset of 22,268 commentary chunks.

Each commentary chunk was ultimately assigned to one of 2 categories— events or others. It is important to note that only the chunks matching to sixes, fours, and dismissals were kept, leading to the creation of highlight that captured the key points of each delivery. This procedure's ability to create a streamlined pipeline for producing cricket highlights serves as evidence of its effectiveness.

YOLOv8			
	F1 Score	Precision	Recall
Bowler Detection	0.97	0.98	0.97
Replay Detection	0.99	0.99	0.99
BERT			
Event	0.97	0.98	0.96
Non Event	0.96	0.96	0.96

TABLE II: Evaluation scores of models used in the proposed architecture.

V. RESULTS AND EVALUATION

YOLOv8's astounding speed and accuracy in real-time object identification have a significant impact on the field

979-8-3503-8132-0/23/$31.00 ©2023 IEEE

of computer vision. Compared to its predecessors, YOLOv8 shows improved accuracy and efficiency as well as the capacity to attain optimal performance with fewer parameters. The trained YOLOv8 model produced remarkable results in the context of our data yielding scores of 98% and 99%, respectively. In Table II, all models' comprehensive evaluation metrics are listed. Notably, the precision and recall ratings displayed by all of our models are excellent. This suggests that our model's efficacy is extremely high anytime it predicts something positive. The F1 scores, which naturally include the weighted harmonic mean of both precision and recall, support the model's performance.

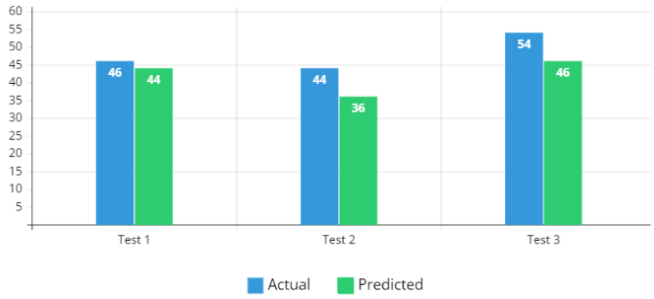

Fig. 6: Correct events predicted by the fine-tuned BERT model are demonstrated here corresponding to each test video. For test 1, 46 actual events occurred and the detected predicted were 44. The remaining test also gave similar results.

It was necessary to evaluate our framework's effectiveness to fully validate it. To do this, three matches that were separated from the training dataset were used for testing. Fig. 6 shows how the complete architecture successfully detected precise events. The procedure included breaking down the balls, extracting the comments, and classifying the commentary. It resulted in the creation of highlights for games lasting five hours, with a processing duration of one hour and twenty minutes on average. Although the performance of our models has been excellent, it is crucial to admit that there have been instances of inferior performance in particular settings. Fig. 7 shows the frequency of all the phrases that lead BERT to classify a ball as either event or non-event.

VI. Limitations

Through the use of the YOLOv8 model, the suggested architecture described in this work exhibits notable capabilities in the context of Bowler Detection and Replay Detection. However, several built-in restrictions have been pointed out within this framework that must be acknowledged. Fig. 8 shows a few such examples.

Firstly, in our case having various camera angles is a must for the effectiveness of the YOLOv8 model. The model's accuracy is noticeably diminished in situations where a single camera viewpoint is used, such as in some games with a fixed straight-angle perspective of the pitch. The lack of different perspectives makes it difficult for the model to accurately identify subtle player actions, which eventually affects its performance.

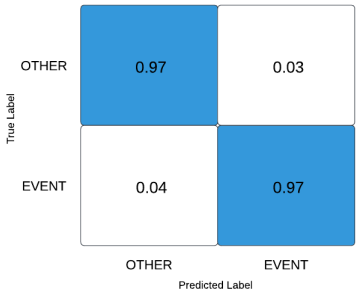

Fig. 7: Confusion matrix of the BERT model shows that 96% of the true 'event' labels were predicted correctly. Similarly, 97% of the true 'other' labels were also predicted correctly. Leaving us with 3% of false positives.

Second, the placement of the scoreboard, which is often at the bottom of the screen, must be recognized by our replay detection system. Changes that move the scoreboard to the upper-right or upper-left corners undermine the system's ability to distinguish between replay sequences. The model's ability to discern replays from live action with precision is undermined by its failure to recognize changes in scoreboard placement effectively.

Fig. 8: In frames A and B, our bowler and replay detection models were able to detect their respective labels. In frame C, the bowler and umpire are conversing but it will be treated as a distinct ball. In frame D, the scorecard is not detected.

Additionally, there are particular difficulties with the Whisper Model, which is used to extract commentary for the first 20 seconds after ball detection. There are times when ball actions come to an abrupt end, which causes a discrepancy between the extracted comments and the real event sequence. For instance, if a boundary is scored on the next ball, the commentary that was supposed to be on the previous ball is actually about the next thing that happened. The commentary's temporal misalignment separates the model's narrative depiction from the real event.

Potential scenarios for misclassification also occur where commentators describe the game's bigger picture, separate

from the actual ball action, such as boundaries and wickets. Unfortunately, the model incorrectly classifies these off-topic conversations as relevant events, adding noise to the event identification process.

VII. Conclusion

This study introduces an impactful architecture for generating Cricket Highlights, incorporating video and commentary text analysis through the adept utilization of YOLOv8 and BERT models, complemented by our CricPulse Dataset. What set cricket apart was its escalating local community engagement and the inherent challenge of its extended match durations. As digital sports engagement evolves, our approach holds promise for reshaping how fans experience and relive the excitement of cricket and beyond.

In the future we plan to expand the CricPulse dataset to enable personalized highlight generation, allowing for tailored selections based on user preferences. Also, the scalability of this architecture to encompass other sports stands as a promising avenue for future exploration, shaping the landscape of dynamic sports content presentation.

References

[1] Vinay Bettadapura, Caroline Pantofaru, and Irfan Essa. Leveraging contextual cues for generating basketball highlights. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 908–917, 2016.

[2] Aman Bhalla, Arpit Ahuja, Pradeep Pant, and Ankush Mittal. A multimodal approach for automatic cricket video summarization. In *2019 6th international conference on signal processing and integrated networks (SPIN)*, pages 146–150. IEEE, 2019.

[3] Tom Decroos, Vladimir Dzyuba, Jan Van Haaren, and Jesse Davis. Predicting soccer highlights from spatio-temporal match event streams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[4] Brian Dwyer, Jared Nelson, Jacob Solawetz, et al. Roboflow, 2022.

[5] Arpan Gupta and Sakthi Balan Muthiah. Learning cricket strokes from spatial and motion visual word sequences. *Multimedia Tools and Applications*, 82(1):1237–1259, 2023.

[6] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.

[7] Abdullah Aman Khan, Yunbo Rao, and Jie Shao. Enet: event based highlight generation network for broadcast sports videos. *Multimedia Systems*, 28(6):2453–2464, 2022.

[8] Maheshkumar H Kolekar and Somnath Sengupta. Bayesian network-based customized highlight generation for broadcast soccer videos. *IEEE Transactions on Broadcasting*, 61(2):195–209, 2015.

[9] Jinyu Li et al. Recent advances in end-to-end automatic speech recognition. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. *European conference on computer vision*, pages 21–37, 2016.

[11] Banoth Thulasya Naik, Mohammad Farukh Hashmi, and Neeraj Dhanraj Bokde. A comprehensive review of computer vision in sports: Open issues, future trends and research directions. *Applied Sciences*, 12(9):4429, 2022.

[12] M Nasir, Ali Javed, Aun Irtaza, Hafiz Malik, and M Mahmood. Event detection and summarization of cricket videos. *Journal of Image and Graphics*, 6(1):27–32, 2018.

[13] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022.

[14] Khushali R Raval and Mahesh M Goyani. A survey on event detection based video summarization for cricket. *Multimedia Tools and Applications*, 81(20):29253–29281, 2022.

[15] Hansa Shingrakhia and Hetal Patel. Cricket video highlight generation methods: A review. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 21(2):1–22, 2022.

[16] Pushkar Shukla, Hemant Sadana, Apaar Bansal, Deepak Verma, Carlos Elmadjian, Balasubramanian Raman, and Matthew Turk. Automatic cricket highlight generation using event-driven and excitement-based features. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1800–1808, 2018.

[17] Khurram Soomro and Amir R Zamir. Action recognition in realistic sports videos. In *Computer vision in sports*, pages 181–208. Springer, 2015.

[18] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer, 2019.

[19] Peng Wang. Research on sports training action recognition based on deep learning. *Scientific Programming*, 2021:1–8, 2021.

[20] Le Xiao and Xiaolin Chen. Enhancing llm with evolutionary fine tuning for news summary generation. *arXiv preprint arXiv:2307.02839*, 2023.

[21] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. A survey of modern deep learning based object detection models. *Digital Signal Processing*, 126:103514, 2022.

[22] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 766–782. Springer, 2016.

979-8-3503-8132-0/23/$31.00 ©2023 IEEE