Proceedings of the International Conference on Sustainable Computing and Smart Systems (ICSCSS 2023)
IEEE Xplore Part Number: CFP23DJ3-ART; ISBN: 979-8-3503-3360-2

Upload Cricket Match Video to Generate Audio Commentary by YOLOv8 and Transformer

[1]KARNATI SAI SHASHANK, [2]N. PRANEETH PRASAD, [3]K. SPOORTHY REDDY,[4]L. SRIDHARA RAO,
Department of Information Technology, J. B. Institute of Engineering and Technology, Hyderabad.
[1]ksaishashankscience@gmail.com [2]praneethprasad148@gmail.com,
[3]spoorthyreddy103@gmail.com, [4]sridhararao.it@jbiet.edu.in

2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS) | 979-8-3503-3360-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSCSS57650.2023.10169522

Abstract— **The main purpose is to post cricket videos and create audio commentary. Make cricket video automatically generate audio commentary. The YOLOv8 model is used to extract the features from the image and is followed by Transformer-LSTM network to generate the response as text, which is then converted to audio. The proposed model serves variable length input data and consecutive outputs. In addition, the model can use timing information for predict the pitch and the length of the bowler's delivery and the batsman's shot selection, and the outcome of the ball. However, there is no standard data to perform those tasks. So, this study performs data collection to classification.**

Keywords— **YOLOv8, Transformer, Bidirectional LSTM, Audio Commentary, OpenCV, movipy, gTTS**

I. INTRODUCTION

Cricket has a massive global audience of over 2.5 billion fans, making it the second international sport watched. The ICC Men's Cricket T20 World Cup 2022 saw a staggering 4.5 billion minutes of viewership on hotstar. The most viewed match was between Pakistan and India. Cricket leagues and tournaments also attract huge numbers of viewers worldwide, offering opportunities for businesses to profit. While certain areas, such as umpiring or commentary, can be automated, generating commentary requires a deep understanding of the game.

In order to generate automatic commentary, it relies on the basic detection of objects, the identification of ongoing activities, and the conversion of those activities into speeches. For nearly a century, television and now the internet have made video streaming of sport and events possible. In sports, verbal and written commentary has always been important, providing viewers and followers with a clear understanding of the game's events. However, individual preferences can lead to biased commentary, making it essential to provide with commentary.

There is a relatively low degree of automation in the production of expert interpretations, but artificial intelligence has been applied for predicting cricket scores, shot classification, calculating winning percentages and developing team strategies. In cricket, however, applications such as computer vision and the use of natural language processing is behind other sports, especially for automatic generation of reports. Using state-of-the-art algorithms, you can develop models that automatically generate commentary for cricket matches. It revolutionizes the way you generate in ball-by-ball analysis and commentary, with access to all data of every match. In this project, we categorized each cricket ball score into a number of categories.

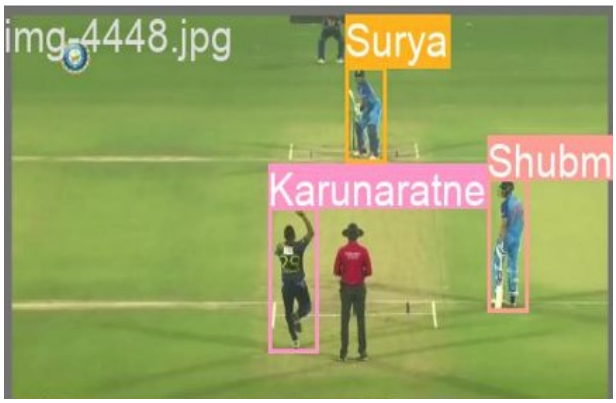

Fig 1. Object detection by YOLO- v8

Using a deep neural network, a single cricket ball video clip serves as input to the model, and by predicting each player's bounding box, we obtain a video with audio commentary as output. This white paper provides a framework for creating upload a ball video for cricket commentary. It can be learned. Using the sequential model approach, we can use words of any length in any size input frame. We also created our own data set containing 234 videos. The proposed framework will be trained and tested on video of invisible crickets using this dataset.

II. LITERATURE REVIEW

Image captioning has capture significant attention, with LSTM-based models producing the best outcomes [1], [2]. However, nowadays we are building AI models on video captioning and description. Sports video caption presents a unique challenge as the model must comprehend the intricate nature of athletics. Prior approaches have involved assigning metadata to videos [3] and organizing captions and videos into clusters. Another method [4] employed a three-stage process where in the semantic components of a sentence were first identified, and a sentence was then generated. During the training phase, the model detected a set of words that could be included in the caption, learning verbs, adjectives and nouns from the image. The next step was to create a text containing the words that the language model recognized. Finally, a deep

multimodal similarity model [4] was used to reprioritize subtitles. Unfortunately, this method did not yield satisfactory results satisficing for generating ball commentary.

Recent research in the area of video captioning has become based on Deep Learning techniques following the success of Neural Networks. Haonan Yu, et al. [5] In order to generate multi-paragraph captions for long videos, a hierarchical recurrent neural network, hRNN, was introduced. A single sentence is not enough for the sports event to be described. It is therefore necessary to provide a few lines of comments. Temporal connections derived from the preceding sentence shall be used in the hRNN model. A sentence generator and a paragraph generator are included in this model.

In these problems we need long term time dependencies and recurrent neural network models. These problems arise from the vanishing and exploding gradient problem, leading to the development of a new model called Long-Short-Term Memory (LSTM) [6]. The LSTM model is an iterative model, although it contains four gates and may store information for a longer period than RNN models.

Researchers use LSTMs to illustrate and describe video content in the literature. Xiang Long et al. [7] Using the Attention layer with LSTM, trained the model to learn important features. In order to anticipate the next word, this model does not rely only on words that have already been used but also takes into account different levels of attention in relation to motion, time and semantic of features.

A. Dilawali, et al [8] Using the Deep Learning system using a very complex computer network that is capable of performing numerous tasks, an algorithm called CNN has been used to identify video content's features from images. In addition, to generate the Natural Language description, an LSTM is used.

Researchers have been focusing on team sports, such as basketball and volleyball, in order to generate captions. Mengshi Qi et al.,[9], generated the subtitles for its volleyball video by a deep framework and an encoding algorithm based on topology LSTM architecture. This approach is characterised by careful representation of movements and the modeling of intragroup relationships.

Before generating commentary for cricket videos, scholars concentrated on classification of frames and annotation issues via deep learning. In order to classify results for each of the ball, which was either Run, Boundary, Six, Wicket, or No Run, Dixit et al. [10] used a Long Term RCC (Recurrent Convolutional Network) and the pretrained VGG16 network model. They pointed out that the position of the cricket ball determines the score or comment on each ball. These are small positions, which can be overlooked in calculations. In the case of AlIslam et al., CNN [11] uses a Convolutional Neural Network to detect cricket players in video clips. For the purpose of exploring commentary creation, these methodologies are a source of inspiration.

Sharma et al. [12] A cricket video annotated with a semantic explanation. Their process involved his two phases, beginning with using information from text comments to classify the video into scenes. We then lined up the frames and phrases and used a classification algorithm to place them into a set of pre-defined classes. Method is effective at annotating videos and classifying them into established categories, but relies on text comments as input.

A video-to-text model using sequence-to-sequence and LSTM proposed by S. Venugopalan [13] aims to teach the system to associate video events with natural language descriptions. The method begins with video decoding by his VGG-16 model [14] applying the CNN approach. And so, he's reading the frames according to their order and using his two LSTM layers is able to generate a description of the video. For commentary on cricket, this is a very good technique. When it comes to the ball's video capabilities, however, there are drawbacks. VGG-16 extracts the predetermined length features, whereas real cricket ball videos have a variety of durations. The model is therefore incapable of handling video with varying length.

The framework proposed by Zain Ul Abideen [15] is based on the encoder/decoder model. Encoder that extracts features from videos. We use a pretrained VGG-16 [14] model to extract features from frames. Each frame extracts fixed-length features (4096). The extracted features are passed to a LSTM based sequence-to-sequence model to generate caption text sequentially. Features that were extracted from a cricket ball video with VGG-16 was an input. Based on features, different lengths of words are generated by estimating conditional probabilities of word sequences. The model doesn't detect the player in the frame, and in generated commentary the model doesn't generate the player's name in commentary. It also extracts only fixed-length features from fixed-length frames. Add zeros if less, remove if more. So, the information is lost, the player is not detected in the frame and the player's name is not generated in the commentary.

Our model is based on transformer bidirectional LSTM (TRANS-BLSTM) which learns to generate text commentary and YOLOv8 to extract features of detected objects happening in a video. First, the video is pass to YOLO-v8 [16], and these reads frames sequentially and after that, extract features of detected objects pass to TRANS-BLSTM to generate commentary of the videos using transformer model. No dataset is available to meet the requirement commentary for cricket ball videos. So, in this paper we are presenting ball-by-ball video commentary of cricket dataset and labelled data in yolo format.

III. Proposed Methodology

A. Methodology

The proposed model is based on the encoder and decoder model. Encoder that extracts features from videos. We extract features from frames using a custom pre-trained YOLO v8 model [16]. All frames from the thrown ball clip are selected, passed through the YOLO v8 model and appended to an array. Extracted features were passed to a Bidirectional Transformer Long Short-Term Memory (LSTM) [6] to generate labels

sequentially. The input is a frame of video, can any number of frames. The output can be any number of different words in the text.

The proposed bidirectional Transformer LSTM model was used in a machine translation problem where natural language inputs are transformed into a natural language output and where length is variable. Natural language inputs are used in machine translation problems, we had used features which are extracted from cricket videos of detected objects using YOLO-v8 as input. We had extracted features by passing bounding boxes of objects detected in each frame into convolution layers and appended into an array. Based on this recognition, we extract features from the recognized objects and generate words of variable length. As explained before, we first get the feature vectors one by one from the input frames. They are then generated sequentially (one by one) using a decoding module containing the LSTM output words. Bidirectional Transformer LSTM (Long Short-Term Memory) is a type of neural network architecture used for sequence-to-sequence learning tasks such as natural language processing, speech recognition, and machine translation. This architecture combines the interactivity of LSTM with the self-awareness mechanism of Transformer.

The formula for the bidirectional LSTM can be written as follows:

$$\begin{aligned} f_g &= \sigma\,[(A_{fb}\times B_{t-1}) + (A_{fc}\times C_t) + b_f] \\ i_g &= \sigma\,[(A_{ib}\times B_{t-1}) + (A_{ic}\times C_t) + b_i] \\ g_g &= \varphi\,[(A_{gb}\times B_{t-1}) + (A_{gc}\times C_t) + b_g] \\ o_g &= \sigma\,[(A_{ob}\times B_{t-1}) + (A_{oc}\times C_t) + b_o] \\ P_g &= f_t\times P_{t-1} + i_g\times g_g \\ H_g &= o_g \tanh(P_g) \end{aligned} \quad (1)$$

Where:

f_g = represents the forget gate
i_g = represents the input gate
o_g = represents the output gate
g_g = cell state at timestamp
P_g = represents candidate for cell state at timestamp
H_g = represents final output
σ = represents the sigmoid function
A_x = weight for respective gate(x) neurons
B_{t-1} =previous LSTM block output
C_t = Current input
b_x = biases for respective gates(x)

The information to be forgotten is determined by the forget gate layer. To make a decision and print an output using the sigmoid function, examine ht1 and xt. The information that will be updated is determined by the input gate layer. The new value should add to the state is provided by Tanh Layer gt. In order to compute a new cell state ct, an old cell state with the designation of ct1 should be modified using gt, tt and ct1. The new cell state formula 2 indicates that the information to be ignored and added are deleted when multiplying a previous cell state by ft. The output gate will use a sigmoid function on the output to take into account xt and ht1. The function of the sigmoid is specified as follows

$$\sigma(i) = 1/(1 + e^{-i}) \quad (2)$$

Output is multiplied by the Tanh function of cell state in order to obtain a hidden state. It is defined as

$$\tanh(i) = (e^{i} - e^{-i}) / (e^{i} + e^{-i}) \quad (3)$$

The formula for the Transformer is specified as follows:

$$\text{Attention}(D,E,F) = \text{softmax}(DE^{T}/\sqrt{d_E})\,F \quad (4)$$

$$\text{Multiheaded}(D,E,F) = \text{Concat}(\text{Attention}(DA^{q}, EA^{K}, FA^{v})) \quad (5)$$

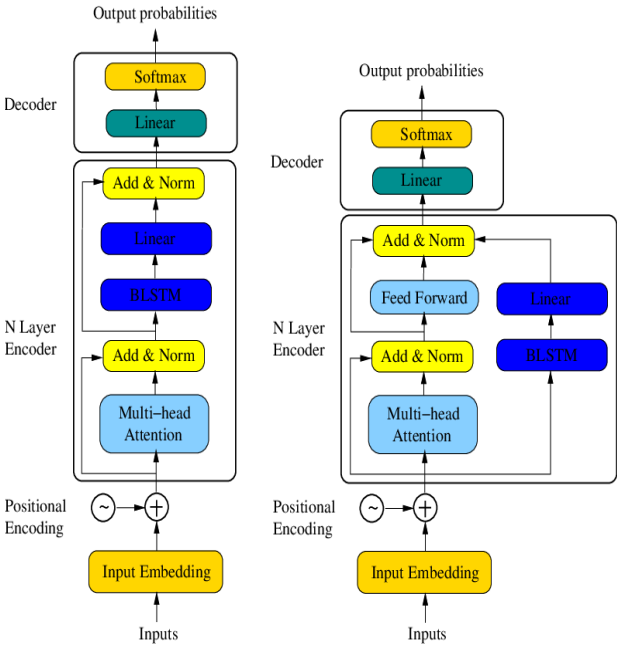

Fig 2. Architecture of Transformer bidirectional LSTM

The Transformer, on the other hand, is a neural network architecture uses self-attention to process sequences of inputs. Self-attention allows network to attend to different parts of the input sequence and to capture long-range dependencies. To combine the bidirectional LSTM with the Transformer, we can use a multi-layer Transformer encoder to process the output of the bidirectional LSTM. The output of each layer is passed to the next layer as input, with the final layer outputting the encoded sequence. The first step is to prepare the text data for the Transformer model. This typically involves tokenization (breaking up the text into individual words or sub words), converting the tokens to numerical vectors, and creating data splits for training, validation, and testing. Next, we added the number of layers, heads, and hidden dimensions, and defining the self-attention and feedforward layers and trained the model

The proposed model is trained on the full match video from the 2023 Ind vs SL T20 match and then converting it to frames with fps=1 using ffmpeg software. Then in the video

of the whole cricket match we got 18450 frames. We labeled all images with bounding boxes of 26 classes (22 players, Wide, Out, Four, Six) in Yolo format. We then trained YOLOv8 on the 26 detection classes and extracted features from the detection objects stored in arrays of variable length. We had built a Transformer Bidirectional LSTM that accepts a variable length input trained on per-ball videos and 5 corresponding commentaries per ball to generate text commentary for the input ball videos. We then used pyttsx3 to convert the text to audio mp3 format and combined the generated audio file with the video with help of moviepy to create a video with an audio commentary of the input video.

IV. Experimental Analysis

The result is a cricket commentary data set containing 234 videos, each with 5 similar commentaries. Each of ball video represents a thrown ball. In addition to the dataset, we have bounding boxes for every frame to identify the player in frame in Yolo format. We then pre-trained YOLOv8 model with custom 26 class. It is detecting accurately. During our experiments, the maximum number of images i.e., frames selected from a video are 1024 frames. Value can be varied and compromises were observed. A high time step value increases memory consumption, and a low time step value with fewer frames can lose of data. Whenever, videos which had frames or images less than of 1024 the left-over frames are not appends with null values i.e., zeros but it selects the same number of frames. For videos has excess frames, it selects only 1024 frames in training. So, model is accepting variable length of input data maximum of 1024 frames. Generates text commentary and converted to audio with detecting players in frame.

A. Dataset

In this research, we create a new dataset for per-ball commentary, as there is no existing dataset available to perform this task. we used T20 game video between Ind and SL. This format ensures 20 overs in each team's innings. In the over he has 6 balls and around 234 balls are bowled. The number may increase wide balls, and may decrease if all batters in the team out early. We split full match video into 234 videos. Each video has information about the ball being thrown and his five commentaries on each video scrapped from the internet [17],[18]. Vocabulary size of commentaries is 114 words. Each video depicting the ball has a set of five comments from him describing the event from the perspective of a bowler and a batsman. It also contains images of the full game video and bounding boxes with 26 classes as labels in Yolo format.

B. Tools and Environment for Experimentation

The model requires a GPU environment with Ultralytics installed on any Python idle to run the YOLO model and Flask to run the web page. To run this model, we need to install OpenCV, pyttsx3 and moviepy.

C. Programming language

It was built using the TensorFlow, Keras, and Ultralytics libraries of the Python language used to build the model. Python is also the most widely used programming language because it is easy to learn and use. There are several open-source libraries to implement machine learning and deep learning models in the Python programming language. TensorFlow is one of these technologies. Keras is an open-source library, provides for artificial neural networks.

Generated Caption: Madushanka to Suryakumar, FOUR runs, bowls a good length it puts into shot for four towards offside
Ground Truth: Madushanka to Suryakumar, FOUR runs Surya ends with a boundary. He sees the field and expects a short wide slower one. He shuffles across, and then puts a lot of whips into the shot to send this over square leg for four.

Fig 3. Results of Generated commentary

D. Result

Cricket ball Commentary Dataset which has ball by ball video, is passed in our model. The commentary generated by our model given when we inputted a ball video.
We had check similarity score for each video by spacy open-source software library and calculated accuracy. We got around 72% accuracy for all 234 videos which is very good similarity score.

E. YOLO-v8

You Only Look Once version 8 algorithm is an Object Detection models from Ultralytics providing state-of-the-art performance. It is built as a unified framework for training Object Detection, Instance Segmentation, and Image Classification models. We had used YOLOv8m for detection with frame size of 640. Right away, YOLOv8 models seem to perform much better compared to the previous VGG16 model and it also detects players in frame. When i valid with test data i got following results.

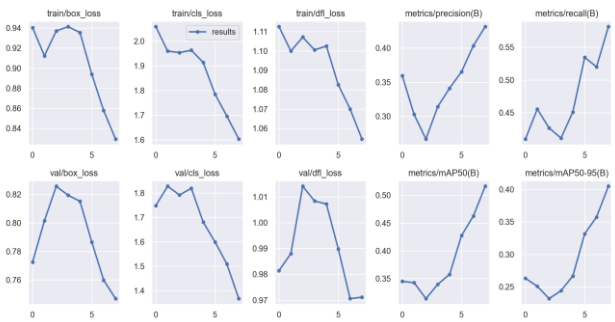

Fig 4. Results which show how better YOLOv8 can detect

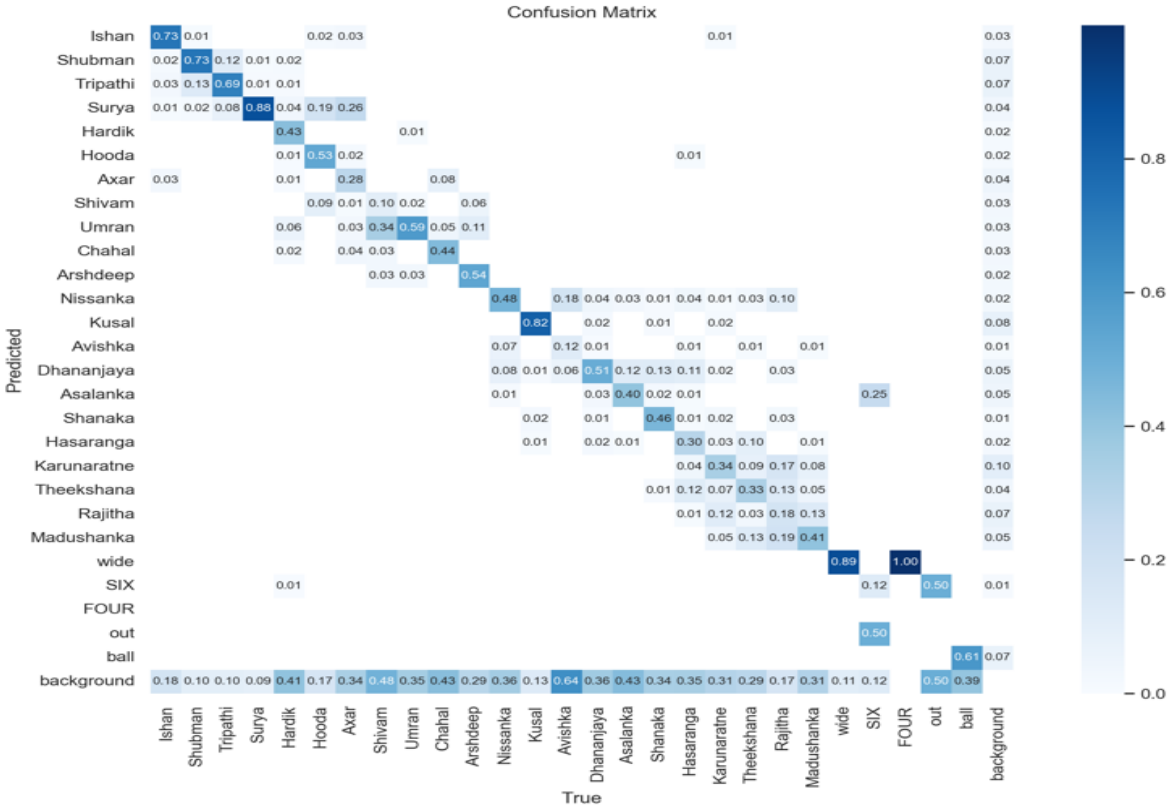

Fig 5. Confusion Matrix of YOLOv8 model which trained

F. Observation

The proposed model allows players to be detected with their name, which was not possible in previous existing works. This model extracts features from the recognized player within the frame rather than the features of the entire frame. So, we can pass the features of player's and the selected shots and shot outcomes to an NLP model. It can now generate understandable and more meaningful commentary that include player names and with past data. It can then be converted to audio format and combined with the cricket ball video.

V. CONCLUSION

This study has presented a novel technique for auto-generating players detection and audio commentary. The proposed model also predicted ball outcomes or the line and length of a thrown ball, we generated an audio commentary. This solution takes cricket ball video clips, extract features using YOLOv8 as input, recognize the player in the frame, and pass it through Transformer's bi-directional LSTM decoder network to generate a cricket ball video commentary. The evaluation results show that the model works and learns timing information for ball toss and hitter shot selection. In the future, it can be used to generates same length of audio for inputted video better than human commentary and detects all cricket players.

REFERENCES

[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3156–3164.

[2] M. Z. Khan, S. Jabeen, M. U. G. Khan, T. Saba, A. Rehmat, A. Rehman, and U. Tariq, "A realistic image generation of face from text description using the fully trained generative adversarial networks," IEEE Access, vol. 9, pp. 1250–1260, 2021.

[3] H. Aradhye, G. Toderici, and J. Yagnik, "Video2text: Learning to annotate video content," pp. 144–151, 2009.

[4] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, "Translating video content to natural language descriptions," pp. 433–440, 2013.

[5] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," 2015.

[6] Zhiheng Huang, Peng Xu, Davis Liang, Ajay Mishra, Bing Xiang, Mar. 2020. [Online]. Available: https://arxiv.org/abs/2003.07000

[7] X. Long, C. Gan, and G. de Melo, "Video captioning with multi-faceted attention," 2016.

[8] A. Dilawari, M. U. G. Khan, A. Farooq, Z. Rehman, S. Rho, and I. Mehmood, "Natural language description of video streams using task specific feature encoding," IEEE Access, vol. 6, pp. 16 639–16 645, 2018.

[9] M. Qi, Y. Wang, A. Li, and J. Luo, "Sports video captioning via attentive motion representation and group relationship modeling," IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 8, pp. 2617–2633, 2020.

[10] K. Dixit and Stanford, "Deep learning using cnns for ball-by-ball outcome classification in sports," 2016.

[11] M. N. A. Islam, T. B. Hassan, and S. K. Khan, "A CNN-based approach to classify cricket bowlers based on their bowling actions," 2019.
[12] R. A. Sharma, P. S. K, and C. Jawahar, "Fine-grain annotation of cricket videos," 2015.
[13] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence – video to text," pp. 4534–4542, 2015.
[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.
[15] Zain Ul Abideen, Saira Jabeen, Summra Saleem, Muhammad Usman Ghani Khan, "Ball-by-Ball Cricket Commentary Generation using Stateful Sequence-to-Sequence Model," in 2021 International Conference on Communication Technologies.
[16] Ultralytics, "Yolo v8 model," Jan 2023. [Online]. Available: https://docs.ultralytics.com/
[17] Cricket commentary, "IND vs SL T20 match", 2023. [Online]. Available: https://www.espncricinfo.com/series/sri-lanka-in-india-2022-23-1348629/india-vs-sri-lanka-3rd-t20i-1348642/full-scorecard
[18] Cricket commentary, "IND vs SL T20 match", 2023. [Online]. Available: https://www.cricbuzz.com/cricket-scores/59960/ind-vs-sl-3rd-t20i-sri-lanka-tour-of-india-2023