2021 International Conference on Communication Technologies (ComTech) | 978-1-6654-1901-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ComTech52583.2021.9616676

Ball-by-Ball Cricket Commentary Generation using Stateful Sequence-to-Sequence Model

Zain Ul Abideen*, Saira Jabeen*, Summra Saleem*, Muhammad Usman Ghani Khan†
*Al-Khwarizmi Institute of Computer Science UET, *Lahore*
{*zain.abideen, saira.jabeen, summra.saleem*}*@kics.edu.pk*
†Computer Science Department UET, *Lahore*
usman.ghani@uet.edu.pk

***Abstract*—Due to the availability of high performance computational devices and enormous video data, deep learning algorithms are assisting for human understandable description of videos. Automatic commentary generation of cricket videos take advantage of aforementioned intelligent techniques. VGG-16 network facilitates extraction of visual pattern from frames followed by encoder-decoder LSTM model. Proposed model can handle variable length input data to output variable number of sequential output. Moreover, the model has ability to encompass temporal information to predict the line and length bowled by bowler, the shot selection of batsman and outcome of the ball. Due to unavailability of cricket commentary dataset, a novel cricket commentary dataset containing video-commentary pairs is presented. Evaluation is also performed on benchmark video captioning datasets which are Microsoft Video Description Dataset (MSVD) and MSR - Video to Text dataset (MSR-VTT). Captions generated by our model are evaluated on video captioning metrics which are METEOR, BLEU, ROGUE_L and CIDEr and outperforms the baseline model.**

***Index Terms*—Video Captioning, Cricket, encoder-decoder, Computer Vision, LSTM, Artificial Intelligence**

I. Introduction

Cricket is the second most watched international sport with more than 2.5 billion fans around the globe [1]. The International Cricket Council (ICC) Men's Cricket World Cup 2019 had 3.5 billion minutes watched across Facebook and YouTube [2]. Most watched match was Pakistan Vs India which had 273 million unique users tuning into TV that day for this match [2]. Cricket live streaming is not just limited to international contests. There are also leagues and tournaments which are watched around the world in huge numbers, such as, Pakistan Super League (PSL) in Pakistan, Indian Premier League (IPL) in India, Bangladesh Premier League (BPL) in Bangladesh, Big Bash League (BBL) in Australia and Caribbean Premier League (CPL) in West Indies.

AI is being used today in cricket for predicting score, shot classification, win percentage calculation, and building strategy for teams. However, applications of computer vision and natural language processing are lagging behind in cricket as compared to other sports. One such area that needs more attention is automatic cricket commentary generation. A model for generating automatic commentary for cricket matches can be developed by using state-of-the-art computer vision and natural language processing (NLP) algorithms. This can be revolutionary as dense commentary and in-depth ball-by-ball analysis can be generated with access to complete database of every match ever played.

Important areas for detailed cricket commentary include line and length of ball that bowler has bowled, shot selection of batsman and outcome of each ball. There has already been work done for each area mentioned independently using classical machine learning algorithms and deep learning techniques for shot classification [3], [4]. Given the challenging nature of the problem, previous work does not address the problem of automatic cricket commentary generation. Deep learning has been applied in different sports for automatic commentary generation. Examples are tennis [5], baseball [6] and volleyball [7] etc.

In this paper, we propose a deep learning methodology for generating ball-by-ball cricket commentary generation. This deep neural network has ability to learn temporal information along with long term dependency using LSTM. Sequential model approach allows variable length of words on variable frames of input. We have also produced a novel dataset containing 240 videos. Our dataset has been made publicly available.[1] Using this dataset, the proposed framework is trained and tested on unseen cricket videos. Furthermore, the model is also trained and tested on Microsoft Video Description Dataset (MSVD) which is a benchmark dataset for video captioning [8]. The generated captions are evaluated using METEOR, BLEU 1-4, ROUGE and CIDEr scores. These metrics compare computer generated captions with human annotated descriptions called ground truth.

This document provides literature survey of video caption-

[1] https://github.com/livezaain/CricketCommentary.

978-1-6654-1901-7/21/$31.00 ©2021 IEEE

ing techniques in Section II. Methodology of our proposed model is explained in Section III. Section IV explains the experimentation as well as the details of our novel dataset and MSVD dataset. Section V provides results of our model on cricket commentary dataset and MSVD dataset. Section VI provides conclusion.

II. Related Work

Captioning of images has received ample attention. LSTM based models gave state of the art results for image captioning [9], [10]. Now the focus is shifting towards video description and captioning. Video captioning for sports is considered a challenging task as the model needs to understand the complex nature of sports. Early work includes tagging video with metadata [11] and clustering the captions and videos. A three-stage method was used by [12] which first identified the semantic part of sentence (subject, object, verb) and then this model generated a sentence based on the template. During this training process, set of words were detected which can be a part of the caption. The model learns nouns, verbs and adjectives from the image regions. In the next step, a language model generates text that includes detected words. In the last step, global deep multimodal similarity model [12] is used to re-rank the captions. This approach does not give satisfactory results in cricket commentary generation as template-based approach gives disjointed descriptions.

After the success of deep neural networks, recent work done on the video captioning applies deep learning models. Haonan Yu *et al.* [13] proposed a hierarchical recurrent neural network (hRNN) model which gave multiple sentence caption to a lengthy video. In sports, an event cannot be captioned using a single sentence, rather, commentary of two to three lines should be used to describe the event. The hRNN model uses the temporal dependencies from the previous sentences. The model has sentence generator and paragraph generator.

In problems where long term temporal dependency is required, the recurrent neural network (RNN) model does not work well. There are vanishing and exploding gradient problems, that is why a new model long-short term memory (LSTM) was proposed [14]. It is also recurrent but has four gates and remembers the information longer than RNN.

In literature, researchers have used LSTM for video captioning and description. Xiang Long [15] *et al.* have used attention mechanism along with LSTM so that the model learns important visual features. This model not only just uses previous word to predict next word but it also uses different attention layers to focus on parts of motion, temporal and semantic features.

A. Dilawari *et al.* [16] proposed deep learning framework which extracts visual features from video using fine-tuned multi-task convolutional neural networks (CNN) and natural language description is generated by using LSTM.

Group activity sports like basketball and volleyball has got researchers attention for caption generation. Mengshi Qi *et al.* [17] used Hierarchical LSTM-based deep framework, which has encoder-decoder architecture, for volleyball video captioning with attentive motion representation and group relationship modeling.

Before working on commentary generation of cricket videos, researchers worked on classification and annotation problems using deep learning. Dixit *et al.* [18] used Long-Term Recurrent Convolutional Network using pre-trained VGG16Net model to classify the outcome of each ball which can be wicket, boundary, run or no run. They pointed out that outcome or commentary of each ball depends on location of cricket ball which is very small and can be lost in the calculations. Al Islam *et al.* [19] has used convolutional neural networks (CNN) to identify bowlers in a cricket video. These approaches provided inspiration to work on commentary generation.

Sharma *et al.* [20] annotated cricket videos with semantic descriptions. They divided this into two stages. First, the video is divided into scenes by using category information from text-commentary. Secondly, the scenes and phrases are aligned and uses machine learning classification algorithms to classify those into a known set of categories. Their approach works well to annotate the videos into known categories, but it relies on text commentary that should be given as input.

Our model takes inspiration from Sequence to Sequence Video to Text model based on LSTM proposed by S. Venugopalan [21] which learns to associate events happening in a video with a natural language description. First, the video is decoded using CNN based model VGG-16 [22], and these frames are read sequentially and after that, description of the videos are generated sequentially using two layers of LSTM.

There are several datasets publicly available for video captioning tasks. J. Xu *et al.* [23] presented MSR-VTT (MSR Video to Text). It is benchmark dataset for natural language description of videos. Das *et al.* [24] presented YouTube videos dataset which are related to cooking and every video has natural language description of the scene. [8] presented MSVD dataset which has videos from YouTube and natural language description of each videos. However, none of the available dataset fulfill the requirement of commentary for cricket videos. That is why novel ball-by-ball cricket commentary dataset is presented in this work.

III. Methodology

The proposed framework is based on an encoder-decoder model. Popular networks available in literature are AlexNet, VGGNet (containing VGG-16 and VGG-19), ResNet and Inception. Encoder is a convolutional neural network (CNN) which extracts visual features from the video. We are using pre-trained VGG-16 [22] model for extracting features from a frame. As the name suggests, VGG-16 has 16 layers with 138 million trainable parameters. Eighty frames from a clip of a ball bowled are selected and passed through VGG-16 model and appended so that the shape of a feature matrix is (80, 4096). The extracted features are then passed to a long short-term memory (LSTM) based sequence to sequence model and captions are generated sequentially. Input is the frames of the video which can be any number of frames. Similarly, output is the varying number of words in a sentence. We do this by estimating conditional probability for words sequence (w_1, …, w_m) given input frames (f_1, …, f_n).

$$p(w_1, \ldots, w_m | f_1, \ldots, f_n) \qquad (1)$$

LSTM model proposed in [14] has been used in machine translation problem in which natural language input is translated into natural language output and their lengths are variable. Instead of natural language input being used in machine translation problem, we use features extracted from cricket video using VGG-16 as input. Based on these features, variable length of words are generated.

As explained, we first get feature vector from input frames of the cricket video one by one. Then using the decoding module comprising LSTM output words are generated sequentially (one by one). This approach solves the problem of variable length of input frames and output words of the sentences,

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) [14] which was designed to learn long term dependencies. It has a chain of repeating cells of neural network. Standard RNN network has one layer in one cell but LSTM has four layers in one cell and they interact with each other in a special way.

$$\begin{aligned}
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)\\
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)\\
g_t &= \phi(W_{xg}x_t + W_{hg}h_{t-1} + b_g)\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \qquad (2)\\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t\\
h_t &= o_t \odot \phi(c_t)
\end{aligned}$$

Forget gate layer decides which information we should forget. It looks at h_{t-1} and x_t to decide and gives output by using sigmoid function. Input gate layer decides which information should be updated. Tanh layer g_t gives new value that should be added to state. The previous cell state denoted by c_{t-1} is now updated by using the i_t, g_t, t_t and c_{t-1} and new cell state c_t is calculated. Equation 2 for new cell state suggests that by multiplying previous cell state with f_t information we should forget is thrown away and new information is added. The output gate looks at x_t and h_{t-1} and gives output by applying sigmoid function. Sigmoid function is defined as

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad (3)$$

To get the hidden state, output is multiplied with tanh of cell state. Tanh function is defined as

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad (4)$$

The detailed description of the proposed model and dataset are given in the following paragraphs.

The working of our proposed model is shown in Figure 1 where input frames are encoded into extracted features vector of fixed size (80,4096) by one LSTM shown in layer 1. When encoding is done second LSTM shown in layer 2 in Figure 1 is used to get the sequence of outputs in natural language. There are two LSTMs in the model as depicted in Figure 1 in layer 1 and 2; then these are unrolled with the passage of time to show how these two encode and decode the frames. Using same LSTMs for encoding and decoding pipeline enables us to have parameters sharing between the two stages and enhance the learning capabilities of the model.

There are 500 hidden units in each LSTM of the model. The arrow from first LSTM shown in layer 1 to second LSTM shown in layer 2 shows the hidden representation h_t from first one is fed as input x_t to second one; so layer 1 handles the input fixed-length extracted features of video frames and layer 2 handles the output sequence of natural language words.

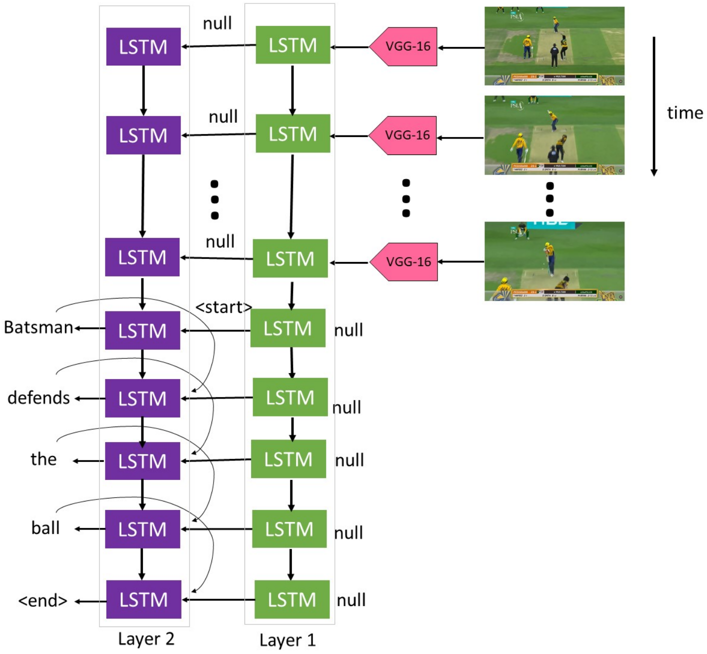


Fig. 1. Architecture of the proposed Model shows that raw frames are passed through VGG-16 for features extraction. These features are passed to encoder LSTM. Second layer outputs words given hidden representation and input text sequentially. Tags $< tart >$ and $< end >$ are used to mark start and end of sentence. Null means vector of zeros is passed

During the encoding phase, the LSTM layer 1 in Figure 1 is fed with the fixed length extracted feature vector of each frame of the cricket video sequentially and during this phase the LSTM layer 2 is fed with the hidden representation and this LSTM concatenate this input as time passes along with null words as input which are actually zeros. When all frames are passed through the LSTM layer 1, the $< start >$ tag is fed to the layer 2 that is when the decoding starts. Now the layer 2 LSTM which has stored the hidden representation now generates the sentence word by word, now the layer 1 of LSTM is fed with null as encoding has ended. After the layer 2 LSTM generates the last word of the sentence; it generates $< end >$ tag which means the natural language generation has now ended. Given the input the hidden representation the model calculates the log-likelihood of output sequence of words. Now this value is maximized and optimized using Adam Optimizer. The loss is not calculated when the model is in encoding phase it is calculated only in decoding phase. The null input in Figure 1 means that input is a vector of zeros.

IV. Experimentation

The Ball-by-Ball Cricket Commentary dataset is presented, which contains 240 videos with three different commentaries for each video. Each video represents a cricket ball bowled. In addition to this dataset, our algorithm is tested on MSR-VTT and MSVD datasets and evaluation metrics were calculated. Input RGB frames of the video are fed into pre-trained VGG-16 model. During the experimentation, the number of frames selected from one video was 80. This value was varied and a trade-off was observed; the memory consumption was increased when we take higher value of time step and lower number of frames for lower value of time step can lose a lot of information. Time step 80 for LSTM layer proved a good value. Using 80 frames where each frame's feature vector has size 4096 output of second last layer of VGG-16 model, batch size of 10 was used during training due to the memory limitations. Whenever, there were videos which have frames less than 80, the remaining frames were padded with null i.e. zeros. For videos of large size, the frames were selected within the size of 80 in training.

A. Datasets

Some of the available datasets for video captioning include MSR-VTT (MSR Video to Text) [23], YouTube videos dataset containing cooking videos [24] and MSVD [8] dataset. However, none of the available datasets fulfill the requirement of commentary for cricket videos. So, a novel dataset containing cricket commentary is presented as well.

1) Microsoft Video Description Dataset (MSVD): In addition to the cricket dataset the algorithm is tested on MSVD dataset [8]. This dataset contains short YouTube videos which shows single activity. Then each video is annotated with a single sentence describing the activity happening in the video.

2) Ball-by-Ball Cricket Commentary Dataset: In cricket, there are different formats of the game. Most popular formats of cricket are Twenty-Twenty (T20) cricket, One Day cricket and Test cricket. Now-a-days, widely played format is T20 cricket as it is short format. In this format, there are 20 overs reserved for each team innings. As there are two teams that play a match, maximum overs to be played are 40; there are 6 balls in one over and nearly 240 balls are bowled in a T20 match. Number of balls can slightly increase due to no balls or wide balls and it can also slightly decrease if all team players get out early. We have divided the whole match into 240 videos where each video contains information of one ball bowled and a text file containing self annotated three or four possible commentaries for each video. The videos come from a cricket match played in Pakistan Super League (PSL) at Dubai International Cricket Stadium in 2018. The vocabulary size of the dataset is 101 words. Corresponding to each video representing a ball, there are 3 commentary sentences describing the events happening with respect to bowler and batsman perspective.

TABLE I
Datasets for the Video Captioning and our dataset

Dataset	MSVD	MSR-VTT	Cricket Dataset
Domain	Open	Open	Cricket
Videos	1970	10,000	240
Sentences	70,028	200,000	527
Words	607,339	1,856,523	4334
Vocabulary size	13,010	29,316	101
Average length	10 sec	20 sec	3 sec
Total length	5.3 hours	41.2 hours	12 minutes

We have focused on the important aspects of the commentary which includes how ball was bowled (e.g. short pitched, bouncer or yorker etc.), its outcome with respect to the shot selected and played by the batsman (e.g. pull, defence or) and which side of the ground ball went (e.g. offside, straight, midwicket etc.). Every commentary line of a ball bowled contains these information.

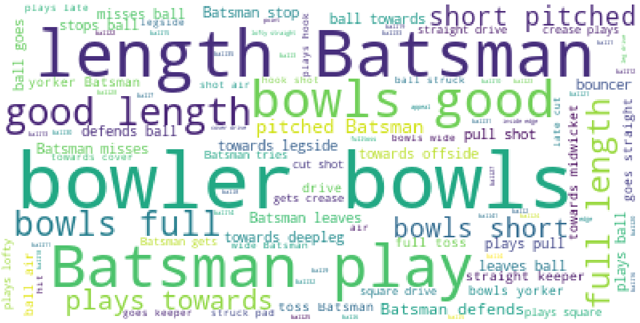


Fig. 2. Word cloud of the proposed ball-by-ball cricket commentary dataset

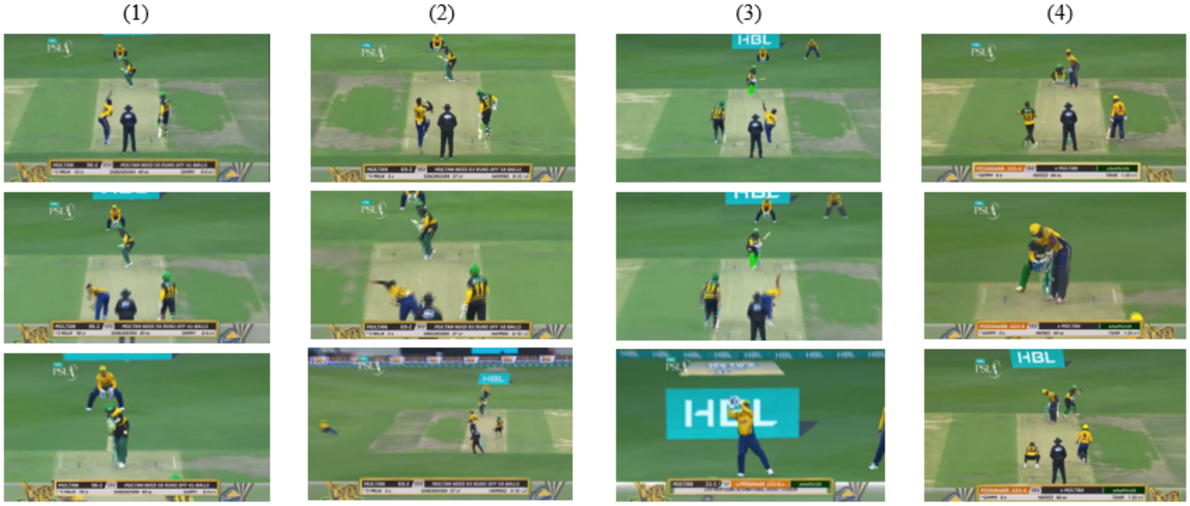
This data is then divided into train and test set with 80% training set and 20% test set.

B. Evaluation Metrics

METEOR [25] metric, which was firstly proposed to evaluate results of machine translation problem, is used to calculate score when a generated sentence and reference sentences are given. The METEOR generates score based on alignment. Using WordNet synonyms, the METEOR checks similar words. Other Evaluation metrics include BLEU [26], ROUGE-L [27], and CIDEr [28].

V. Results and Discussion

Ball-by-Ball Cricket Commentary Dataset which contains ball by ball commentary, is passed in our end-to-end model. The captions generated by our model is evaluated with the evaluation metrics given in Table 2. The METEOR score of 41.9% is very good. One reason for high METEOR score is the reason that the dataset contains relatively lower vocabulary size as cricket commentary words repeat in most of the occasions. For example, batsman, bowler, ball, length etc. are used repeatedly.



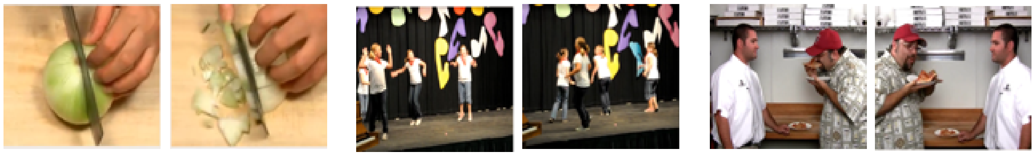
Generated Caption: bowler bowls good-length Batsman defends the ball
Ground Truth: bowler bowls good-length Batsman stops the ball

Generated Caption: bowler bowls good-length Batsman plays towards offside
Ground Truth: bowler bowls good-length Batsman plays towards offside

Generated Caption: bowler bowls short-pitched Batsman leaves the ball ball
Ground Truth: bowler bowls bouncer Batsman leaves the ball and it goes straight to keeper

Generated Caption: bowler bowls full-length Batsman Batsman the ball
Ground Truth: bowler bowls full-length Batsman plays the inside edge

Fig. 3. Results of Commentary generation are shown for four balls bowled. (1), (2) and (3) shows best results, model describes correct bowling information and outcome (4) shows partial results where model has successfully described the bowling information, however, failed to predict the outcome



Generated Caption: A woman is slicing an onion

Ground Truth: A woman demonstrates how to slice an onion

Generated Caption: A group of people are dancing on stage

Ground Truth: A group of children are dancing on stage

Generated Caption: A man is talking

Ground Truth: A man is eating a pizza

Fig. 4. Results of our model on MSVD dataset (1) and (2) shows the best results of MSVD test videos and (3) depicts how model can be confused in a situation where two people are facing each other and that's how two people talk, but in video one person is eating pizza and is not talking

TABLE II
RESULTS OF OUR PROPOSED MODEL ON BALL-BY-BALL CRICKET COMMENTARY DATASET, MSVD AND MSR-VTT

	Our Dataset	MSVD	MSR-VTT
Evaluation Metric	Score (%)	Score (%)	Score (%)
METEOR	41.9	32.2	26.5
BLEU 1	75.4	77.7	76.4
BLEU 2	64.4	64.9	60.1
BLEU 3	53.0	54.2	52.5
BLEU 4	44.4	55.5	43.8
ROGUE_L	74.2	73.7	61.1
CIDEr	91.7	89.7	48.2

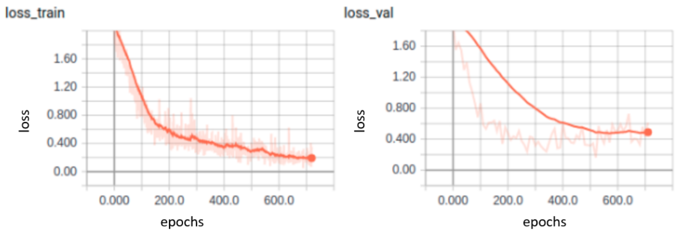


Fig. 5. Training loss and validation loss on cricket commentary dataset

Figure 5 shows the loss as a function of epochs for training data on the left side and validation data on the right side for the cricket commentary dataset.

The results for MSVD and MSR-VTT using our model are also shown in Table 2. Our results are close to the state of the art results where as BLEU 4 results are better than state-of-the-art results [29], [30]. The objective of this research was to generate cricket commentary. Since there was no benchmark dataset available for commentary generation, hence the model was tested on benchmark video captioning datasets of open domains.

Figure 3 shows results of some of the balls from bally-by-ball cricket commentary dataset. In the left most example the model has predicted everything right and some of the snapshots of the video are shown in the column. The first image shows the bowler is about to bowl. In the second image the information about the length of the ball that bowler has bowled is given. This ball has good length. The model can learn the information about the length of the ball by the location of the ball. Since the size of ball is very small; most of the time that information is lost. But this information can be learnt from the response of the batsman. For example, there are different shots that can be played when a yorker is bowled as compared to the type of shots played when short pitched ball is bowled. Batsman mostly defends the yorker and plays pull short when short pitched ball is bowled.

VI. Conclusion

This paper presents an encoder-decoder based technique for automatic cricket commentary generation. Instead of predicting outcome of ball or predicting information about line and length of the ball bowled, we generated cricket commentary using latent space model. This unified solution takes video clip of a cricket ball bowled as input extracts features using VGG-16 and passes to LSTM decoder network for caption generation of cricket events. The evaluation scores show that the model is working and is learning the temporal information of ball bowled and shot selection of the batsman. Cricket commentary generation benefits visually impaired community. It can be used to generate better commentary than human commentators in the future.

VII. Acknowledgment

Special thanks to National Center for Artificial Intelligence (NCAI) for funding this research work. We would like to express our gratitude to our organization (KICS) for their support.

References

[1] S. Das, "Top 10 most popular sports in the world [updated 2019] - sportsshow.net," sportsshow.net, 08 2019. [Online]. Available: https://sportsshow.net/top-10-most-popular-sports-in-the-world/

[2] I. , "2019 men's cricket world cup most watched ever," www.icc-cricket.com, 09 2019. [Online]. Available: https://www.icc-cricket.com/media-releases/1346930

[3] D. Karmaker, A. Z. M. E. Chowdhury, M. S. U. Miah, M. A. Imran, and M. H. Rahman, "Cricket shot classification using motion vector," in *2015 Second International Conference on Computing Technology and Information Management (ICCTIM)*, 2015, pp. 125–129.

[4] M. N. Al Islam, T. B. Hassan, and S. K. Khan, "A cnn-based approach to classify cricket bowlers based on their bowling actions," in *2019 IEEE International Conference on Signal Processing, Information, Communication Systems (SPICSCON)*, 2019, pp. 130–134.

[5] F. Yan, K. Mikolajczyk, and J. Kittler, "Generating commentaries for tennis videos," pp. 2658–2663, 2016.

[6] B. J. Kim and Y. S. Choi, "Automatic baseball commentary generation using deep learning," p. 1056–1065, 2020. [Online]. Available: https://doi.org/10.1145/3341105.3374063

[7] 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.

[8] D. Chen and W. Dolan, "Collecting highly parallel data for paraphrase evaluation." Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 190–200. [Online]. Available: https://www.aclweb.org/anthology/P11-1020

[9] 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.

[10] 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.

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

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

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

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

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

[16] 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.

[17] 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.

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

[19] 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.

[20] R. A. Sharma, P. S. K, and C. Jawahar, "Fine-grain annotation of cricket videos," 2015.

[21] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence – video to text," pp. 4534–4542, 2015.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[23] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5288–5296.

[24] P. Das, C. Xu, R. F. Doell, and J. J. Corso, "A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2634–2641.

[25] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language." Baltimore, Maryland, USA: Association for Computational Linguistics, Jun. 2014, pp. 376–380. [Online]. Available: https://www.aclweb.org/anthology/W14-3348

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," ser. ACL '02. USA: Association for Computational Linguistics, 2002, p. 311–318. [Online]. Available: https://doi.org/10.3115/1073083.1073135

[27] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://www.aclweb.org/anthology/W04-1013

[28] R. Vedantam, C. L. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," 2014.

[29] "Papers with code - msvd benchmark (video captioning)." [Online]. Available: https://paperswithcode.com/sota/video-captioning-on-msvd-1

[30] "Papers with code - msr-vtt benchmark (video captioning)." [Online]. Available: https://paperswithcode.com/sota/video-captioning-on-msr-vtt-1