SpecTextor: End-to-End Attention-based Mechanism for Dense Text Generation in Sports Journalism

Indrajeet Ghosh[†*], Matthew Ivler[‡*], Sreenivasan Ramasamy Ramamurthy[†§], Nirmalya Roy[†§]
[†§]Mobile Pervasive & Sensor Computing Lab, [§]Center for Real-time Distributed Sensing and Autonomy (CARDS)
[†§]Department of Information Systems, University of Maryland Baltimore County, United States
[‡]Department of Computer Science, Pomona College, United States
{*indrajeetghosh, rsreeni1, nroy*}*@umbc.edu*[†], *mnia2018@mymail.pomona.edu*[‡]

2022 IEEE International Conference on Smart Computing (SMARTCOMP) | 978-1-6654-8152-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/SMARTCOMP55677.2022.00081

***Abstract*—Language-guided smart systems can help to design next-generation human-machine interactive applications. The dense text description is one of the research areas where systems learn the semantic knowledge and visual features of each video frame and map them to describe the video's most relevant subjects and events. In this paper, we consider untrimmed sports videos as our case study. Generating dense descriptions in the sports domain to supplement journalistic works without relying on commentators and experts requires more investigation. Motivated by this, we propose an end-to-end automated text-generator, *SpecTextor*, that learns the semantic features from untrimmed videos of sports games and generates associated descriptive texts. The proposed approach considers the video as a sequence of frames and sequentially generates words. After splitting videos into frames, we use a pre-trained VGG-16 model for feature extraction and encoding the video frames. With these encoded frames, we posit a *Long Short-Term Memory (LSTM) based attention-decoder* pipeline that leverages soft-attention mechanism to map the semantic features with relevant textual descriptions to generate the explanation of the game. Because developing a comprehensive description of the game warrants training on a set of dense time-stamped captions, we leverage two available public datasets: *ActivityNet Captions* and *Microsoft Video Description*. In addition, we utilized two different decoding algorithms: *beam search* and *greedy search* and computed two evaluation metrics: *BLEU* and *METEOR* scores.**

***Index Terms*—Sports Journalism, Semantic Knowledge, LSTM, Soft-Attention Mechanism, Beam Search, Greedy Search, Human-Machine Interactive Applications**

I. Introduction

With the influx of the information era, different disciplines are looking to capitalize on the capacities of data and technology; sports are no exception. Though analysts have been tracking team and player statistics for decades, the development of new technologies has reshaped what data is collected, how that data is collected, and what applications it can be used for. The growing applications of sports analytics range from player tracking as a means to increase player performance to annotation of sports clips [1], [2]. There are a number of underdeveloped areas in sports analytics, including sports journalism, which is still heavily dependent on the observations and explanations of a game from experts and commentators. Nevertheless, the field continues to make strides and reshape various sports at all levels of play.

[*]These authors contributed equally to this work

In a similar trajectory, the information era has also seen the rise of machine learning techniques, and with it a growing intersection between the fields of Computer Vision (CV) and Natural Language Processing (NLP). This intersection includes tasks such as video/image captioning that aim to take in a visual input and utilize natural language to describe different features or actions from that input. Adding complexity to the task of video captioning, numerous works attempt to create dense captions. The difference between dense and traditional captioning arises from the specificity and level of detail present in each. Where a traditional video caption may succinctly explain the events of a scene, almost as if to summarize it, a dense caption aims to describe all of the individual actions within a series of events.

Due to the fast-paced nature of sports, dense description is required to get a thorough understanding of the game. Previous works related to sports specific captioning tend to generate dense captions, for dense captions provide an opportunity for a more in-depth explanation of the events in the game. In [3], [4] both emphasize dense and fine-grained captions to elaborate upon sports video. Though both take a multimodal approach to generating viable and detailed captions, [4] applies a soft attention mechanism prior to the LSTM layers in the decoding module. In [5] developed this form of attention which has been widely adopted as a means for increasing video/image captioning. Similarly, in [2] demonstrates a multi event-level approach to developing explanations of sports videos. That said, [2], just like many other captioning works, does not aim to develop dense captions; rather, it annotates the sports videos with distinguishing events that set individual actions apart from each other. These works build upon a set of CV-NLP tasks, in which models translate the video to a varying length text format and demonstrate the merits of CV in sports.

In this work, we aim to create a framework that can understand video of sports games and generate dense text descriptions about the game's actions to aid the field of sports journalism. Furthermore, we postulate an end-to-end attention approach centered application: generating dense sports captioning for aiding sports journalists from gameplay footage.

Below are the overall contributions of this work:

- *SpecTextor: an end-to-end encoder-decoder framework with a soft attention mechanism for the generation of dense captions. The soft-attention mechanism captures*

2693-8340/22/$31.00 ©2022 IEEE
DOI 10.1109/SMARTCOMP55677.2022.00081

the region of interest (ROI) from each frame. We train this model to create dense, event-heavy gameplay explanations within sports videos.

- *We employed ActivityNet Captions and MSVD datasets, two publicly available dense and non-dense captioning for the subjects (activities, objects, sports, etc.). The motivation for employing the MSVD dataset is to demonstrate the generalizability and robustness of the SpecTextor framework as it covers a vast range of subjects and for ActivityNet dataset, solely considered sports-based activities videos. We evaluate using two decoding algorithms, greedy and beam search, on two evaluation metrics: BLEU (1-4) and METEOR (Metric for Evaluation for Translation with Explicit Ordering) scores.*

The remaining of the paper follows the following order. In Section II, we discuss developments and applications related to the field of sports analytics as well as numerous works related to natural language processing in human activity and sports video analytics. Section III explains our model architecture and the various modules implemented. We expand upon our methodology in Section IV, and we explain both the preprocessing phase and experimental setup to test our model in Section V. Finally, we discuss and enumerate the results of our various benchmark metrics in Section VI and discuss the rationale behind our results before concluding and future work in Sections VII and VIII, respectively.

II. Related Works

This section highlights the related work applied methodologies into two categories: sports analytics and natural language processing in human activity and sports video analytics. We mainly focus on summarizing the difference between the existing approaches to the proposed framework.

A. Sports Analytics (SA)

Recent developments in machine learning have made visual input, a readily available form of data, powerful for in-depth sports analysis. The following works demonstrate the various merits and applications of machine learning principles within the discipline of sports analytics. Accuracy and error estimation are sports principles that can elevate a player's performance in the game. In [6], analyzes the accuracy by which an individual performs a particular movement. The work extracts visual features from images and develops an associated scoring metric for evaluating these features in contrast to the target pose. In addition to form analysis, teams and players alike benefit from understanding and mapping player movements. In [1] tracks the positions of players in indoor sports and develops an application for player level movement statistics based on computer vision, template matching, and partitioning algorithms. In [7] though specific to squash, leverages broadcast video inputs and pose estimation/computer vision to track the kinematics of players. Due to developments in motion analysis, teams and individuals can improve their performance by learning from in-depth analysis of games.

Expanding on the performance benefits of machine learning and CV in sports analytics, we note health benefits and automation applications from computer vision in sports. Player safety is a major concern of sports institutions, and [8] combines VGG-19 features and wearable sensor data to construct a framework for posture analysis with applications to reducing injuries caused by high-risk postures. Building off of a desire for automation in sports analytics, in [2] introduces a three-level framework dependent on time-specific video segments, frame-level object detection, and frame-level pose modeling to develop annotations of racket sports from gameplay footage automatically. Our work aims to address the sub-discipline of sports journalism and aid journalists in sports analysis via dense descriptions of sports games from game footage.

B. Natural language processing in human activity and sports video analytics

In this section, we look at past works in video captioning. Recently, numerous papers have developed architectures and methods for natural video captioning.

Though we do not take a multimodal approach, a number of multimodal studies use mechanisms and datasets similar to our own. [9] tests a novel multimodal fusion mechanism based on attention, taking audio as well as visual inputs from video. The domain of multimodal captioning [4], emphasizes volleyball video captioning and implements an encoder-decoder architecture with soft-attention similar to our own but requires multimodal input data consisting of pose modeling, trajectory mapping, and group relationships extracted from videos. In addition, [4] trains and evaluates on MSVD, ActivityNet Captions, MSR-VTT [10], and Sports Video Captioning Dataset-Volleyball (SVCDV), a novel dataset made for the paper. [3] similarly uses a multimodal approach, accounting for skeleton modeling, costly pixel-by-pixel segmentation, and relationship modeling between players. These modalities are parsed from convolutional neural networks but encoded using multiple sets of LSTMs. These papers generate dense captions on sports videos, similar to our work but with different inputs and architectures. In addition, we are employing the same datasets, MSVD and ActivityNet Captions datasets.

Unimodal works also provide insights into methods for captioning. In [11] proposes using hierarchical reinforcement learning on unimodal visual inputs to generate fine-grained descriptions. Aligning more similarly with our architecture, [12], though not focused on sports or dense/fine-grained captioning, utilizes a single modality to extract frame features. While [12] uses InceptionResNet-v2 as the pre-trained feature extraction model, we utilize VGG-16. Considering these various architectures and tasks, we utilize the unimodal input of semantic representations following by LSTMs layers with a soft-attention decoder to provide a novel application for creating dense sports descriptions.

III. Overall Architecture

In this section, we will enumerate and discuss the overall architecture of the *SpecTextor* framework. As seen in Figure

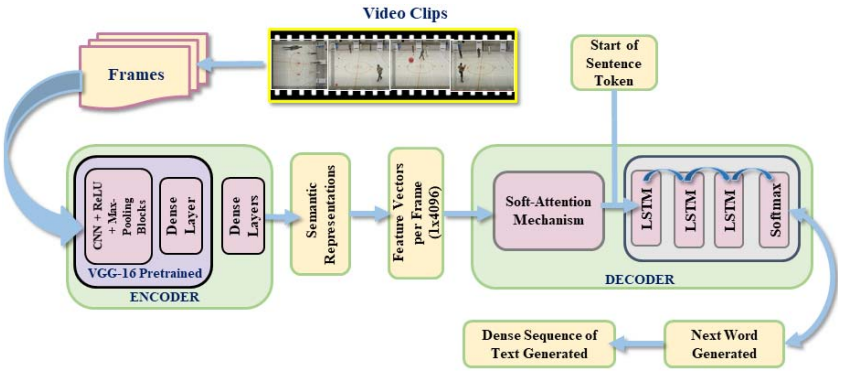

Fig. 1: An end-to-end automated text generator constitutes the encoder and soft attention-based decoder modules. We considered VGG-16 pre-trained architecture on the ImageNet dataset for the feature extraction module. Moreover, we considered dense and three LSTM layers for the encoder and decoder modules, respectively.

1, we use an encoder-decoder based architecture. Our encoder takes in a sequence of frames and processes them through a pre-trained VGG-16 model. This model consists of numerous convolutional neural networks followed by ReLU operations and max-pooling blocks. This feature vector is passed through the second part of the encoder, a trainable, fully connected layer. This layer allows the encoder to learn during back-propagation without retraining the parameters of the VGG-16 model. This layer presents us with semantic representations of the various video frames.

We utilize this set of semantic representations from each frame in the soft attention mechanism of the decoder. For each iteration of the LSTM block, we feed in the one-hot encoding index of the input word. This one-hot encoding is then manifested as a word embedding. The output of the soft attention mechanism is concatenated to this embedding vector prior to being run through the three LSTM layers in the LSTM block. Per iteration of the LSTM, the soft attention mechanism takes in the entire set of semantic representations and the decoder's hidden state to develop its output. Upon ending one iteration of the LSTM block, the decoder has produced a hidden state matrix and cell state matrix, as well as a vector of the vocabulary size in which each element refers to the likelihood of each word in the vocabulary, is the next word. This vector is used to determine the next word, which then provides new input for the next iteration of the decoder. Furthermore, the initial input to the LSTM block is a "Start of Sentence" (SOS) token, and the final output should be an "End of Sentence" (EOS) token. However, it will also stop if it reaches the maximum caption length, which varies depending on the dataset.

IV. Methodology

In this section, we discuss and highlight the overall proposed pipeline. We discuss the uses of the different components of the architecture as well as what each piece provides and its role in the translation from video to captions. We discuss the three primary modules for feature extraction, attention/ROI weighting, and sequence generation.

A. Encoder:Feature Extraction Module

The encoder consists of two primary components. First is the pre-trained VGG-16 model. Utilizing a well-known pre-trained feature extraction model such as VGG-16 allows for faster model adjustment, consistent feature outputs, and faster training. VGG-16 pipeline outputs a 4096 element vector per input frame. These vectors are passed to a trainable dense layer. This dense layer utilizes the feature vectors output by VGG-16 and adjusts the values to fit the model's needs. Because we do not retrain VGG-16, the final dense layer, post feature extraction, is crucial for translating the features into the optimal form for the decoder.

B. Soft-Attention Module

The soft attention module is a part of the decoder module. It takes in the features extracted from the encoder as well as the current decoder hidden state. The hidden state provides a guide for what features within the frames to emphasize for evaluation in the LSTM layers. This module satisfies the need for cohesion from word to word, adding weight to particular features based on previously generated words.

Mathematically, we computed the soft-attention score by assuming the weighted features for the LSTM be: x_1, x_2, x_3, x_4,...,x_n and each denotes a sub-section of an image or a frame. To compute a attention score S_i to measure how much attention for x_i, we assumed context/hidden states from LSTM layers as $C = h_{t-1}$.

$$S_i = tanh(W_c C + W_x x_i) = tanh(W_c h_{t-1} + W_x x_i) \quad (1)$$

We pass S_i to a softmax layer for normalization to compute weights α_i, where α_i softmax (S_1, S_2, S_3, S_n). with softmax, α_i, adds up to 1. We computed a weighted average for each features (x_1, x_2, x_3, x_4,...x_n)

$$Z = \sum^{i} a_i x_i \quad (2)$$

At last, we will use Z in place of x as the LSTM features.

C. LSTM-based Decoder Module

The RNN-based decoder takes in a concatenated vector of attention-applied features and the embedded version of the input token. It processes this input in a 3-layers of LSTM. The motivation for employing three layers of LSTMs is because it obtains a better semantic knowledge and also reduces the computational complexity and time required to produce the "scene descriptions" for each frame. It determines both the hidden states for the next iteration and a probability vector representing the likelihood that each word in the vocabulary is the following word in the output sequence. The hidden states are passed into the next iteration of the decoder, but the probability vector is used to determine the generated sequence of words. These output vectors are compared to the ground truth words during training to calculate the loss and update the model via back-propagation. However, we utilize two different methods to determine which sequence to output as the final predicted sequence in evaluation. In one method, we utilize a Greedy Search decoding method. The highest probability index from the output vector is taken as the prediction and translated into its one-hot encoded corresponding word. This method repeats generating the output sequence word by word. Alternatively, we use the Beam Search method for decoding the LSTM output. This requires considering conditional probabilities of entire sequences and picking the sequence with the highest conditional probability where the candidates for the sequences are determined to be a set of k candidates who have the highest conditional probability at each LSTM iteration. This cycle continues for the maximum length of the caption and determines the final sequence upon completion.

For generation one word at a time, we start the text with an "SOS" token:

$$Next\ Word = f(image, last\ word) \tag{3}$$

Mathematically, replace the image x in LSTM decoder model:

$$h_t = f(last\ word, x, h_{t-1}) \tag{4}$$

X is the frame or image and h_t is the hidden state to predict the next word at the time step t. Furthermore, we continue until we predict the "EOS" token. As the soft attention we will replace the generalized frame or image with a more attention on region-of-interest (ROI).

$$Next\ Word = g(h_t) \tag{5}$$

$$h_t = f(last\ word, attention(x, h_{t-1}), h_{t-1}) \tag{6}$$

The attention module generates the most important features representing the ROI in the frame/image.

V. Experimentation Pipeline

The experiments were conducted on a Linux server which housed an Intel i7-6850K CPU, 4x NVIDIA GeForce GTX 1080Ti GPUs and 64GB RAM. All the codes for data preprocessing and deep learning mechanisms were implemented with Python. Deep learning pipelines were implemented using PyTorch libraries. In addition, we used OpenCV and Matplotlib libraries in Python for frame retrieval and plotting. Because of the various sizes of the clips between the two datasets, we used 28 frames per MSVD video and 100 frames per ActivityNet Captions video. The overall framework constitutes 3 LSTM layers with 256 memory cells with a dropout ratio of 0.1 for each layer. In addition, we employed an $Adam$ optimizer with a learning rate equal to 0.0001 with a batch size of 16.

A. Dataset Description

- **ActivityNet Captions Dataset** [13] [14]: is introduced for dense captioning sequential events and actions. It contains 27,801 videos with 849 hours and more than 100k sentences collected from the ActivityNet dataset. The average length of videos is $\approx$ 10 minutes, and also each sentence covers a unique segment of the video to describe multiple events concurrently. Each sentence has an average length of 13.48 words, following a relatively normal distribution throughout the dataset. For the experimentation, we select a subset from the ActivityNet dataset, which is comprised of sports activities videos* and split it into training, validation and testing sets of 300, 70 and 30 videos, respectively, from the various sports.
- **Microsoft Video Description Dataset (MSVD)** [15]: contains a pool of 1,970 short videos collected from YouTube. Each video describes a single activity in a wide range of subjects (actions, sports, etc.). The dataset comprises 80,839 sentences, and each sentence has about 8 words. We select 1,200 videos as the training set, 100 for validation and 670 as the testing set following the same setting employed in [15].

B. Data Pre-processing

For the ActivityNet Captions dataset, we only desired the subset of videos related to sports. We created a list of sports-related videos from the dataset. For each video that we used, we extracted an evenly distributed subset of frames from the video. The feature matrix of each video has the same dimensions, and the frames are evenly spread throughout the video. Because of this frame sampling method, we can keep down the computational cost of running the model while ensuring we do not only take a clip from the start of the video. Because MSVD was used to test generalizability, we utilized all of the available videos. We took a subset of frames from each video, and for each frame, we extracted features using the VGG-16 extraction model. This leaves us with a features vector for that frame. After doing this for all of the frames, we stack the vectors into a matrix of ordered feature vectors.

In addition to getting these feature vectors, we also need to set up the vocabulary and ground truth captions. The captions associated with all of the videos in the training set of the datasets are tokenized. Each token is given a corresponding index. This one-hot encoding of the various tokens from the training caption makes up the indices of the output vectors of the proposed model. In addition to setting up the vocabulary,

*http://activity-net.org/explore.html

TABLE I: Evaluation Metrics Scores

Decoding Search	Dataset	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR
Greedy Search	***MSVD***	0.68913	0.53379	0.39986	0.35979	0.27821
	ActivityNet Captions	0.41434	0.35467	0.23980	0.18907	0.20439
Beam Search	***MSVD***	0.64330	0.51154	0.44768	0.39972	0.29768
	ActivityNet Captions	0.40678	0.33089	0.26098	0.20876	0.21967

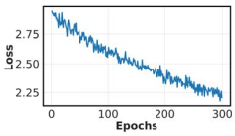

Fig. 2: ActivityNet Captions dataset training Loss

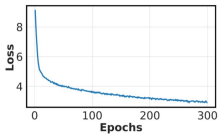

Fig. 3: MSVD dataset training Loss

Fig. 4: GENERATED: A man is shooting on a gun
TARGET: A man is firing two weapons

Fig. 5: GENERATED: A cat is playing with a dog
TARGET: A tortoise is playing with a cat

we set up the ground truth captions for testing. This is simply a matter of taking them from the original dataset and setting them up in the dataloader as the associated caption (or set of captions in the case of MSVD) for a particular features matrix.

C. Evaluation Metrics

For the evaluation of the *SpecTextor* framework, we utilized two evaluation metrics: BLEU (Bilingual evaluation understudy) shown in equation 7 which evaluate the closeness/precision of the machine translation to human reference translation, where BP: brevity penalty, N: No. of n-grams, we usually use unigram, bigram, 3-gram, 4-gram, w_n: Weight for each modified precision, P_n Modified precision. Another evaluation metric: METEOR (metric for evaluation for translation with explicit ordering) shown in equation 10, checks sentence alignment and word matching. It modifies the precision (P) and recall (R) by replacing the recall with a weighted F-score based on mapping unigram (m), c is the number of matching chunks and penalty (Pe) for incorrect word order shown below:

$$BLEU = BP * exp(\sum_{N}^{n=1} w_n \log p_n) \tag{7}$$

$$\textit{Weighted F-Score} = \frac{PR}{\alpha P + (1-\alpha)R} \tag{8}$$

$$\textit{Penalty (Pe)} = \gamma(\frac{c}{m})^{\beta}, \textit{where } 0 \leqslant \gamma \leqslant 1 \tag{9}$$

$$\textit{METEOR} = (1 - \textit{Penalty}) * \textit{Weighted F-Score} \tag{10}$$

VI. Results and Discussions

This section highlights and enumerates our insights and findings from the *SpecTextor* framework. Firstly, we utilize the popular BLEU (1-4) and METEOR metrics to evaluate the overall performance. The results of these metrics categorized by two decoding search types and datasets are in Table I. A higher score for each of these metrics demonstrates a better performance of the proposed framework in generating a caption close to the ground truth. In this table, we see two significant trends. First, we notice that the MSVD dataset results have higher scores than the ActivityNet Captions dataset across all our experimental settings, regardless of the search algorithm. Second, Greedy Search algorithms perform better than Beam Search on BLEU-1 and BLEU-2, but not on BLEU-3, BLEU-4, and METEOR. From this, we glean two significant occurring phenomena. Our first point denotes that the proposed framework works better when generating smaller captions. This demonstrates its potential for application or adjustment toward providing better dense captions. Our second point presents more information about the application of the different decoding algorithms. On the one hand, Greedy Search performed better on the lower n-gram BLEU metrics, denoting it likely has better word selection. However, the higher n-gram BLEU and METEOR scores emphasize ordering and coherence. Such phenomena occurred due to the Beam Search considering a number of candidate sequences and choosing the optimal one based on conditional probability. The probability of a given the word is heavily dependent on the rest of the sentence. This is in contrast to the Greedy Search, which will pick the most likely word in the decoder output at a given time. This means that Greedy Search will have a better word selection. This is a benefit for metrics like BLEU-1 or BLEU-2, which only check if two sentences have the same words or word pairs, but it is also a detriment when a sentence needs proper ordering. Because the Beam Search will consider this ordering, it may pick a particular word that does not have the maximum probability in the decoder output to gain a larger conditional probability for the sequence. Thus, it is essential to use Beam Search rather than Greedy Search to obtain the best caption as a coherent sequence of words.

Upon further evaluation of the *Spectextor* performance, we see various learning trends as well as potential reasons for non-

Fig. 6: GENERATED: A man is playing with a rope
TARGET: This man is playing with his pet dog

competitive evaluation scores, even on the more generalized and traditionally captioned MSVD dataset. Figures 2 and 3, show us the loss per epoch of the models during training on each of the datasets. While MSVD dataset training sees a steep learning curve early on that eventually evens out to a more gradual rate, the ActivityNet Captions dataset sees a constant downward trend with heavy fluctuation throughout. Regardless of evaluation metrics, the models were learning and approaching an optimal generation over time. In addition, Figures 4, 5, and 6 show examples of captions generated and target captions provided on videos from the MSVD dataset. Among these, we get a better understanding of the above results. Though the model trained on the MSVD dataset tends to get the right actions and can appropriately capture the general events of a clip, it tends to have issues with properly identifying particular features. In the case of Figure 5, that manifests as a tortoise being confused with a dog, and in Figure 6, a fluffy dog being confused for rope. From this, we conclude that a more robust semantic-based pre-trained encoder architecture could improve the feature extraction and results of the model.

VII. Conclusion

In this paper, we propose an end to end attention-based dense text generation framework for generating dense captions of sports videos to aid in sports journalism and other human activities. We utilized an encoder-decoder-based framework comprised of a soft-attention mechanism with LSTM layers and two decoding techniques: *beam* and *greedy* search. We experimented on two public datasets: *ActivityNet Captions* and *MSVD* and achieves *BLEU* and *METEOR* scores of **0.64330** and **0.29768**, respectively. Furthermore, we consider two experiment settings (i) select a subset from the ActivityNet dataset, which is comprised of sports activities videos (ii) consider short videos with captions that covers a vast range of subjects (actions, human, sports, etc.). *SpecTextor* enables capturing temporal semantic features with relevant textual descriptions of the subjects. Lastly, we assert that both models were optimizing and learning during training. Some features are misrepresented, even among the MSVD dataset, impeding high accuracy.

VIII. Limitations and Future Work

We would like to highlight a few limitations and the scope of the future work. Firstly, we would like to collect our in-house dataset with more robust and dense captions than the state-of-the-art datasets with multiple possible captions for ground truths aimed at the specific events better suited for sports journalism and other subjects. In addition, ActivityNet is itself a generic dataset that only emphasizes particular parts of a video that may not be what a journalist deems the most important details of an event. Furthermore, using more frames from each video would yield more temporal features and thus better input for dense captioning. It would also be advantageous to leverage the stronger traditional captioning of the architecture by adding a module that divides a longer video into multiple event-specific clips. This would allow for the model to use more frames from clips deemed as relevant and ignore unimportant clips. These shorter clips would take advantage of the model's success on the MSVD dataset. Finally, we believe the VGG-16 model may be misinterpreting the feature representation of the videos. We want to investigate more sophisticated and robust pre-trained models for better feature representation and minute discrepancy: bidirectional transformers, multi-head attention mechanisms, etc., to yield optimal and high-performance for the human-machine interactive applications.

IX. Acknowledgement

This research is supported by the NSF Research Experience for Undergraduates (REU) grant # CNS-2050999, NSF CAREER Award # 1750936 and U.S. Army Grant # W911NF2120076. In addition, the authors would like to thank all our program affiliated dignitaries and graduate students for their valuable feedback.

References

[1] E. Monier, P. Wilhelm, and U. Rückert, "A computer vision based tracking system for indoor team sports," *In The fourth international conference on intelligent computing and information systems*, 2009.

[2] D. Deng, J. Wu, J. Wang, Y. Wu, X. Xie, Z. Zhou, H. Zhang, X. Zhang, and Y. Wu, "Eventanchor: Reducing human interactions in event annotation of racket sports videos," pp. 1–13, 2021.

[3] H. Yu, S. Cheng, B. Ni, M. Wang, J. Zhang, and X. Yang, "Fine-grained video captioning for sports narrative," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6006–6015, 2018.

[4] 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, 2019.

[5] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *International conference on machine learning*, pp. 2048–2057, 2015.

[6] B. Jia, "Recognition model of sports athletes' wrong actions based on computer vision," 2020.

[7] M. M. Baclig, N. Ergezinger, Q. Mei, M. Gül, S. Adeeb, and L. Westover, "A deep learning and computer vision based multi-player tracker for squash," *Applied Sciences*, vol. 10, no. 24, p. 8793.

[8] D. Sharma and V. Sharma, "Novel architecture for reducing sports injuries in football," *International journal of convergence in healthcare*, vol. 1, no. 1, pp. 10–16, 2021.

[9] C. Hori, T. Hori, T.-Y. Lee, K. Sumi, J. Hershey, and T. Marks, "Attention-based multimodal fusion for video description." IEEE, 2017, pp. 4193–4202.

[10] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5288–5296.

[11] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang, "Video captioning via hierarchical reinforcement learning." IEEE, 2018, pp. 4213–4222.

[12] S. Olivastri, G. Singh, and F. Cuzzolin, "End-to-end video captioning." IEEE, 2019, pp. 0–0.

[13] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, "Dense-captioning events in videos," in *International Conference on Computer Vision (ICCV)*, 2017.

[14] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2015, pp. 961–970.

[15] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko, "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2712–2719.