2021 IEEE 7th World Forum on Internet of Things (WF-IoT) | 978-1-6654-4431-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/WF-IoT51360.2021.9595891

Towards Optimized IoT-based Context-aware Video Content Analysis Framework

Gad Gad
School of Engineering and Applied Sciences
Nile university
Giza, Egypt
g.gad@nu.edu.eg

Eyad Gad
School of Engineering and Applied Sciences
Nile university
Giza, Egypt
e.gad@nu.edu.eg

Bassem Mokhtar
College of Information Technology
University of Fujairah, Fujairah, UAE
Faculty of Engineering
Alexandria University, Egypt
bassem@uof.ac.ae

***Abstract*— Despite the success of convolutional neural networks (CNNs) in the area of spatial analysis, and recurrent neural networks (RNNs) on sequence modeling and interpretation tasks, video analysis has only seen limited interest and progress. This is partially due to focusing on the natural humanlike translation from video space to natural language space to the detriment of informativeness. This paper is proposing an automated context-aware video analysis framework that is directed by the constrains of its application. This framework encorporates an encoder-decoder neural network trained on a closed-domain video-to-text dataset. The network architecture and the standardized language model present in the dataset are optimized for speed, to allow the system to be applied on IoT devices, and for informativeness, to extract information easily from the model output to the following stages of the anlaysis. The proposed framework provides a practical method to integrate the power of CNN and RNN combination in a directed way to extract the most from video content. A classroom monitoring system is discussed as an example of the capabilities and limitations of the proposed framework using NVIDIA's Jetson nano board.**

Keywords—Context-aware Analysis, Video Description, Closed-domain Dataset, Subject-Verb-Object Description, Deep Learning Framework.

I. Introduction

Video analysis is the task of extracting information from an input video. One branch of this task that has a lot of attention already is video captioning which focuses on generating natural language from an input video. While generating natural language makes sense when the target user is human, it doesn't fit in applications where the generated sequence is part of an automated pipeline and thus is processed after it is generated to extract information.

Current approachs addressing video describtion are mostly based on an encoder-decoder architeture combining convolutional neural networks (CNNs), for spatial encoding, and recurent neural netwokks (RNNs), for temporal encoding and decoding. These methods focus mainly on training neural networks to perform a seemless translation between different modalities: video, text, and possibly audio. this is evinced from the convention of training such methods on massive datasets of pairs of open-domain videos to open-domain describtions, namely the MPII Movie Description corpus (MPII-MD) [1], the Microsoft Video Description dataset (MSVD) [2], the Montreal Video Annotation Dataset (M-VAD) [3], and the Microsoft Research Video to Text (MSR-VTT) [4]**.** the latter two datasets have been annotated using Amazon Mechanical Turk (AMT) which suggests an inconsistent sentence structure of annotations which in turn works against standardization of the output. In addition, evaluating these methods on the same evaluation metrics used for language modeling and translation problems.

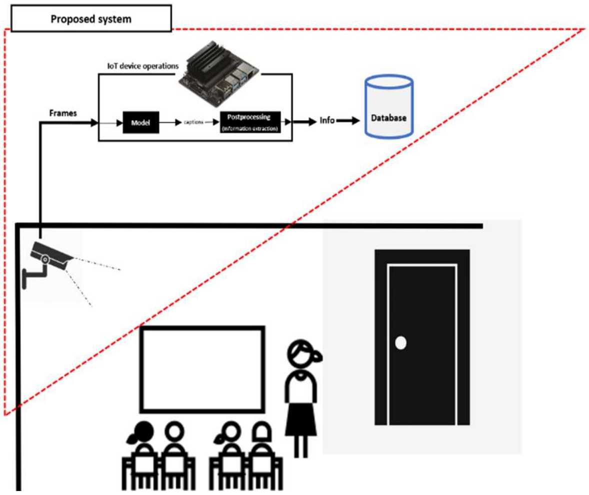

Figure 1 Overview of the proposed system in a class monitoring setup.

Limiting the video analysis problem to video describtion limits the amount of structured information that can be derived from a rich content like videos by distributing CNNs' feature extraction capabilities over the large volume and complex dynamics of open-domain videos and focusing RNNs' language modeling power to learn the language model in the dataset. With edge devices' limited computing capabilities, . Moreover, the generated description requires further interpretation for most applications involving automation to extract meaningful data.

This paper is proposing a mutli-step framework which comprises some of the current video describing components to perform video analysis on a particular application. The performed analysis is optimized in the sense that:

1- Limiting the sentences in the dataset to closed-domain vocabulary related to a particular application reduces the training effort and learning complexity on RNNs. Also, using predefined sentence structures makes extracting information from the generated sequence much easier yet allow the system to be extended and learn to analyze new behaviors and event.
2- Limiting the videos in the dataset to closed-domain videos related to a specific appliaction reduces the complexity of the content and improves the performance of CNNs to extract relevant features.
3- Reducing the complexity of the dataset results in reducing the complexity of the model used thus

allowing it to be used on IoT devices for applications like servaillance and indoor automated management of classrooms or warehouses.

The proposed system is then applied on an IoT device to be tested in real-time. Figure 1 shows an overview of the setup of the experiment.

This paper's remain is organized as follows: Section II explores the related work done in the area of video description. The proposed approach for building a directed video analysis system is discussed in secion III. Experiment setup on IoT device and evalution is presented in section IV. Finally, the paper is concluded in Section V.

II. Related work

Recent work on video description is inspired by the advances in sequence to sequence models like neural machine translation (NMT) [5], speech recognition [6], and image captioning [7] all of which take an end-to-end approach of an encoder-decoder architecture that encodes the input sentence (or image) into embeddings that are then fed to a recurrent neural network (RNNs) of gated recurrent units (GRUs) or long short term memory units (LSTMs) decoder which generates the output sequence. While different variants of RNNs can be used to encode and decode sequences, RNNs and GRUs do not perform well on long-dependecies. LSTMs have proven to yeild better results.

When applied on video description, LSTMs are used to encode the sequence of embeddings, that were generated from applying a CNN feature extractor on frames, into a single vector that encodes all the video's relevant information to generate text. This sequence is then considered the input to another decoder LSTM units that generates the description word by word. For example, [8] incorporates LSTM to descirbe cooking videos. [9] takes a more general approach by LSTM to map a sequence of frames to a sequence of words from a set of YouTube videos from MPII-MD [1] and M-VAD [10]. Finally, [11] calculates the embeddings of each frame in the input sequence using a CNN feature extractor. Then pools these embeddings in a single vector. Then an LSTM decodes this vector to a sequence of words. The two main disadvantages of this approach are losing temporal data by missing the order of the frames after pooling them, and losing spatial data by over compressing the whole sequence of frames in a single embedding vector.

The introduction of the self-attention mechanism [12] further increased confidense in this approach by improving performance on long dependencies in the input sequence. The attention mechanism recalculates the input vectors giving more weights to the most relevant features in the input sequence to generate each prediction in the output sequence. Effectively, attention solves the problem of deschronization between the input and the output sequences, for example, the 1st frame of the video doesn't correspond to the first word in the output description.

Zhou et al [13] built on top of the idea of attention, also known as Transformer, and proposed an end-to-end method to produce video event proposals and captions simulatenously to allow direct influence of the language model to the video event proposal.

In the field of video captioning, some methods [10], [14] consider additional cues to videos, like audio. This is referred to as mutli-modal. The model presented by [14] is a weakly supervised mutli-modal video captioning model. Hessel et al [10] used adds automatic speech recognition (ASR) to improve performance and used a transformer architecture to encode both inputs, video frames and speech tokens and generate captions. Hessel et al approach was limited to instructional (cooking) videos. In addition, Lashin et al [15] proposed a general mutli-modal architecture that can be scaled to mutliple modals and tested it with audio, speech tokens, and video frames on open-domain dataset.

Our proposed system is close to the method described in [16] except that they do not extract a complete Subject-Verb-Object (SVO) description. Our method: (1) extracts SVO descriptions from open-domain datasets. (2) reduces vocabulary size to decrease the complexity and size of the model to allow to run on edge devices. (3) Apply the system on an AI-enabled IoT device and validate its end-to-end inference time cost.

III. approach

An overview of the proposed system is shown in Figure 1, where an IoT-connected camera monitors the classroom and processes the collected frames in to generate a description of the scene in a predefined template. The generated caption is further processed to extract names, dates, and entities based on the metadata (context) given to the system. Finally, the information is inserted in a database.

A. Dataset

As shown in Figure 2, we start by filtering the dataset to include only the video-description pairs that are relevant to the application at hand as well as adding new custom pairs.

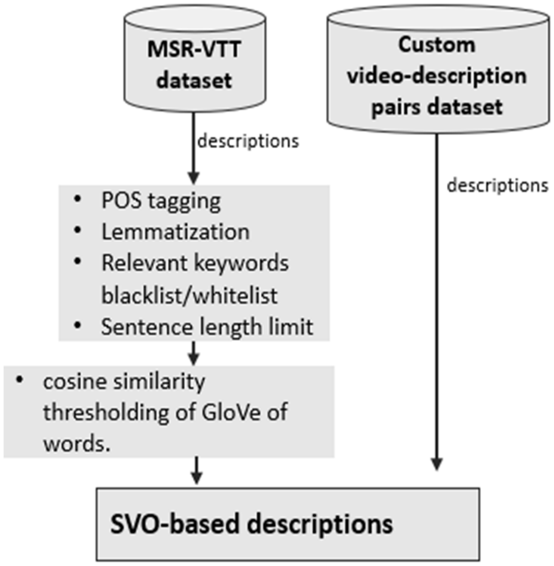
MSR-VTT dataset
Custom video-description pairs dataset
descriptions
descriptions
POS tagging
Lemmatization
Relevant keywords blacklist/whitelist
Sentence length limit
cosine similarity thresholding of GloVe of words.
SVO-based descriptions

Figure 2 Steps for preparing the MSR-VTT dataset and custom dataset from extracting only videos relevant to the application and captions that follow Subject-Verb-Object (SVO) template.

The dataset used is MSR-VTT which contains 10k video clips and 200k total descriptions, 20 captions for each video. Although the video clips in MSR-VTT are categorized, none of its categories match our application (classroom management). We used regular expressions to search for descriptions that include a whitelist of keywords like "class', "student", "teacher" and to exclude a blacklist of keywords which are associated with data that misinterpret the meaning of the word "class" as "car class" for example. Then, we apply POS tagging to discard adjectives, pronouns, adverbs, pre-determiners, etc. and only keep nouns and verbs. After that, lemmatization is applied to return the base form of every noun and verb to make it easier later to calculate frequency of words and compare distance between words. Finally, if the length of the noun-verb combination of a description is 3 and the order is "noun-verb-noun", there is a high probability that this is the SVO format, if not, it can manually be adjusted to the SVO format.

The next step is to reduce vocabulary size to reduce the complexity of the language model the system will have to deal with, this process is done on nouns and verbs separately. All unique words are converted to their GloVe [17] representation, which is a meaningful representation for words where the distance between two-words reflect their semantic relation. The method to reduce the vocabulary size is to cluster close words together by measuring the distance between their GloVe representations and applying a threshold. Figure 3 (a) shows the 2D plot of the PCA of the Glove representation of nouns. Three distance metrics were tried with manual threshold tuning:

$$cosine\ distance = 1 - \frac{u.v}{\|u\|_2 \|v\|_2} \tag{1}$$

$$correlation\ = 1 - \frac{(u-\overline{u}).(v-\overline{v})}{\|(u-\overline{u})\|_2 \|(v-\overline{v})\|_2} \tag{2}$$

$$euclidean\ distance = (\ \sum(w_i |(u_i - v_i)|^2)^{1/2} \tag{3}$$

Where u and v are the two input 1-D arrays, $\overline{u}$ and $\overline{v}$ are the means of the elements of u and v, respectively, and w is a 1-D weights array. After comparing the generated clusters from each distance algorithm to what is expected, Euclidean distance is used because it was found to yield relatively better results. Figure 3 (b) shows the clustered nouns. Words in a cluster are replaced by the most frequent word in that cluster to reduce vocabulary size and simplify the language model.

B. Model

Figure 4 shows the neural network model architecture as well as a flow diagram showing the sequence of operations starting from the input frames and the input caption, the so far generated tokens, and the output token. Starting from top left, the input sequence of frames, one by one, is fed to a CNN feature extractor to generate a sequence of vectors that encode frames' relevant features.

A dual-branch model takes in the vectors generated by the CNNs and, in the other input branch, the sequence of indices of words generated so far. Despite that every activity is represented by only one caption and each caption in strictly 3 tokens (subject, verb, and object), the caption input branch takes 8 tokens. This is due to two reasons: (1) the model allows to detect two separate activities. (2) the special tokens "***seq_start***" and "***seq_end***" must be included at the beginning and ending of the caption. For example, the description "a teacher writes on a chalkboard" is processed as in Figure 5. Many deep learning frameworks offer a collection of models pre-trained on large datasets to be used either directly as a classifier or integrated in a larger model as a pre-processing step (our case). We tried different CNN architectures pre-trained on ImageNet [18]. Since the CNN model is only needed to encode the features of each frame in a representative vector, it can be separated from the rest of the model balance between performance and time cost of the generated vector.

Algorithm 1 – Caption generation from video.

```
Input: frames (list of frames),
       max_len (max length of the generated description sentence),
       word2ix (a dictionary mapping each token to index),
       ix2word (a dictionary mapping each generated index to token),
       model (trained model)
Output: caption (generated sentence)
caption ← [seq_start]
vectors ← [ ]
for (i = 1 to length(frames)) do
    vectors ← add CNN (frames[i])
for (i = 1 to max_len) do
    token_seq ← [ ]
    for (i = 1 to length(caption)) do
        token_seq ← add word2ix (caption[i])
    props ← model.predict (vectors, token_seq)
    index ← argmax (props)
    token ← ix2word (index)
    caption ← add token
    if (token = seq_end) then
        break
return caption
```

C. Postprocessing

After generating the caption from the input video, we map the tokens in the caption to a corresponding function that interprets this function. Different tokens correspond to different functions depending on the required operation for that token. Some tokens, mostly verbs, do not require post-processing and thus are considered final and are not mapped to a post-processing function. One example of a post-processing function is face recognition. A camera facing the classroom door can set the region of interest to the door and interpret the word "teacher" in the sentence "teacher enters class" to the actual name of the teacher using an off-the-shelf face detection and face recognition functions. Words that can be interpreted with the same functions are "student", "man", and "person", replacing all these words from the vocabulary by a general word like "person" reduces the complexity of mapping tokens to post-processing functions, which is done manually.

IV. Experiment and Results

To validate the proposed method, we used the Jetson nano, an AI-enabled IoT board manufactured and supported by $NVIDIA$®. It has a 128-core Maxwell GPU and a Quad-core ARM A57 @1.43 GHZ CPU. The board has two power modes depending on the capability of the power source: $5W$ and $MAXN$ modes. We tested the performance our solution in both modes. To make the best use of $NVIDIA$'s GPU to speed up our model inference, we used $NVIDIA\ JetPack$ an SDK that includes Linux Driver Package (L4T), CUDA-X accelerated libraries and APIs for AI applications.

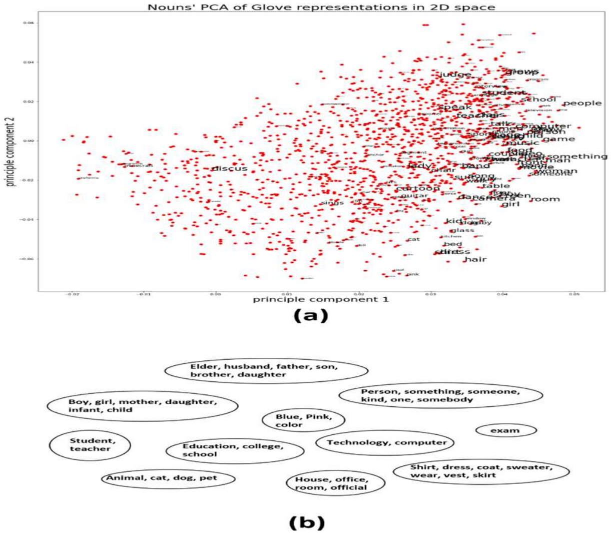

Figure 3 (a) 2D plot of principle components of the GloVe representations of the nouns in the dataset. (b) The clusters formed by applying a threshold on the Euclidean distance between the most frequent verbs and all others.

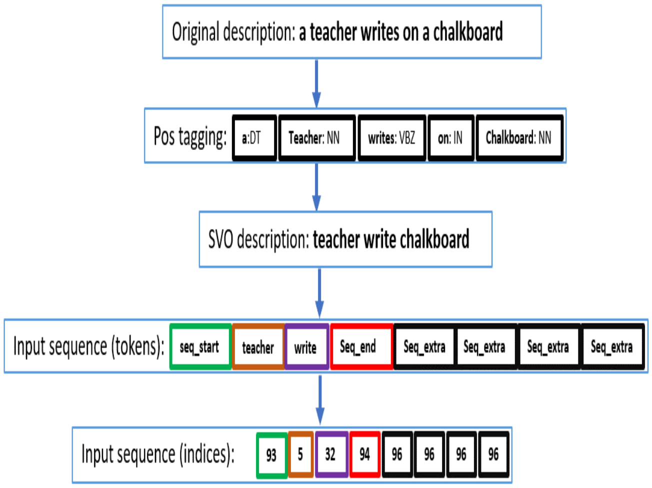

Figure 5 An overview of description processing from tokenization to POS tagging to adding special tokens i.e. seq_start, seq_end, .. to generate the input sequence of indices.

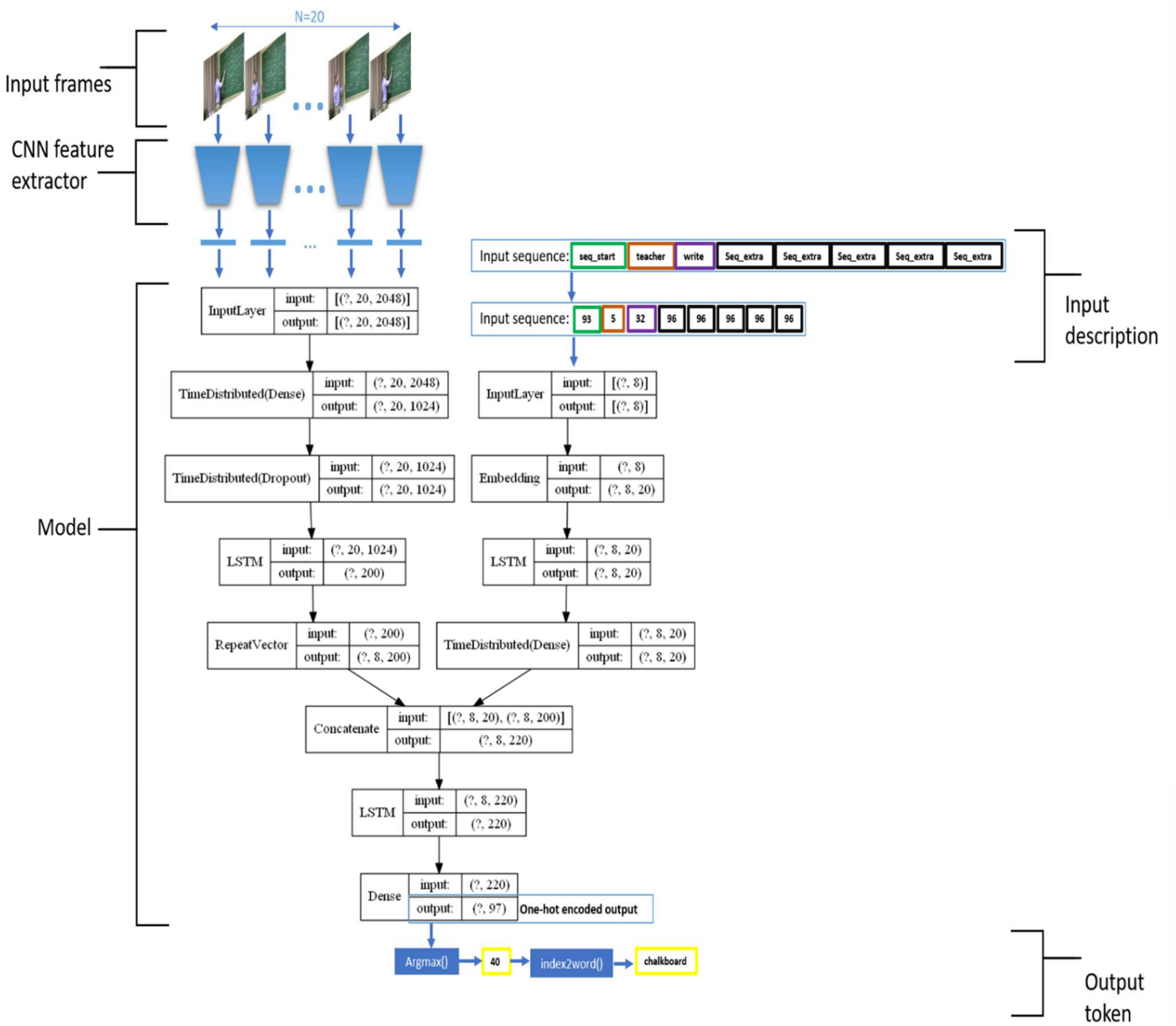

Figure 4 Overview of caption generation process showing the model architecture and the operations applied on the input sequence of frames.

Table 1 shows the time cost of a single inference on three CNN architectures: GoogLeNet [19], ResNet-18 [20], and ResNet-50 on both power modes as well as the time for generating a single caption token (word), under the two power modes.

Table 1 Time cost (Sec) of an inference (CNN) and generating a token.

Phase	**Model**	**MAXN**	**5W**
CNN feature extractor	GoogLeNet	0.33 ± 0.003 s	0.49 ± 0.009 s
	RESNET-18	0.65 ± 0.006 s	1.09 ± 0.007 s
	RESNET-50	$1.166 \pm .304$ s	$1.536 \pm .056$ s
Token generation	—	0.21 ± 0.01 s	0.3 ± 0.02 s

To generate a description sentence composed of len_{tokens} words from a sequence of frames of length len_{frames}, we employ *algorithm 1. The Total Time cost* (TT) *of this process is realized in Eq.4.*

$$TT = (len_{tokens} * 0.21) + 0.33 \tag{4}$$

The previous equation is not dependent on len_{frames} because at any time t the system uses the $len_{frames} - 1$ embedding vectors processed in previous steps and needs to apply CNN only on the current frame at time t.

We measured the accuracy of the generated captions to be **37%**. Given that we could mine only 48 case-related videos from MSR-VTT using the method in figure 2, this is an acceptable accuracy. However, to practically employ this system in any application, a dedicated team should collect reasonable dataset with SVO captions.

V. CONCLUSION

In this paper, we presented an end-to-end framework for light-weight video analysis that is optimized to run on IoT devices. The proposed system starts by automatically generating SVO templates from a given set of captions. We also proposed a method to reduce the vocabulary used in captions by clustering similar words based on the distance between words' embeddings and replace each cluster with its most frequent word. A combination of CNN feature extractor, to extract vector representations from frames sequence, and LSTMs, for caption generation, was used. Furthermore, the system can use the context (meta-data) of the application to map ambiguous words to a set of post-processing functions to extract more useful information. Finally, we validated our framework by applying it to an AI-enabled IoT device (Jetson Nano) which demonstrated an acceptable speed by reaching 0.21 S to generate a single token (word) from the caption of a given sequence of frames embeddings. Our future work includes gathering more data and integrating transformer blocks to the model to improve accuracy.

VI. REFERENCES

[1] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A dataset for Movie Description," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07-12-June-2015, doi: 10.1109/CVPR.2015.7298940.

[2] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, vol. 1.

[3] A. Torabi, C. Pal, H. Larochelle, and A. Courville, "Using Descriptive Video Services to Create a Large Data Source for Video Annotation Research," Mar. 2015.

[4] 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 Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-December, doi: 10.1109/CVPR.2016.571.

[5] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, doi: 10.3115/v1/d14-1179.

[6] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016, vol. 2016-May, doi: 10.1109/ICASSP.2016.7472618.

[7] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07-12-June-2015, doi: 10.1109/CVPR.2015.7298935.

[8] J. Donahue, K. Saenko, T. Darrell, U. T. Austin, U. Lowell, and U. C. Berkeley, "Long-term Recurrent Convolution Networks for Visual Recognition and Description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, 2017.

[9] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence - Video to text," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, vol. 2015 Inter, pp. 4534–4542, doi: 10.1109/ICCV.2015.515.

[10] J. Hessel, B. Pang, Z. Zhu, and R. Soricut, "A case study on combining ASR and visual features for generating instructional video captions," in *CoNLL 2019 - 23rd Conference on Computational Natural Language Learning, Proceedings of the Conference*, 2019, doi: 10.18653/v1/k19-1039.

[11] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *NAACL HLT 2015 - 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 2015, doi: 10.3115/v1/n15-1173.

[12] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, Jun. 2017, vol. 2017-December, pp. 5999–6009.

[13] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-End Dense Video Captioning with Masked Transformer," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, doi: 10.1109/CVPR.2018.00911.

[14] T. Rahman, B. Xu, and L. Sigal, "Watch, listen and tell: Multi-modal weakly supervised dense event captioning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, vol. 2019-October, doi: 10.1109/ICCV.2019.00900.

[15] V. Iashin and E. Rahtu, "Multi-modal dense video captioning," *arXiv*. 2020.

[16] "Improving video activity recognition using object recognition and text mining | Proceedings of the 20th European Conference on Artificial Intelligence."

[17] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, doi: 10.3115/v1/d14-1162.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2010, doi: 10.1109/cvpr.2009.5206848.

[19] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07-12-June-2015, doi: 10.1109/CVPR.2015.7298594.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-December, doi: 10.1109/CVPR.2016.90.