A mobile device framework for video captioning using multimodal neural networks

Rafael J. P. Damaceno* and Roberto M. Cesar Jr.*
Institute of Mathematics and Statistics, University of São Paulo,
São Paulo, SP, Brazil
Email: rafael.damaceno@ime.usp.br, rmcesar@usp.br

***Abstract*—Video captioning is a computer vision task aimed at providing textual descriptions for videos. There are numerous strategies and datasets that can be employed to create models capable of addressing this task. In this study, we have devised a deep learning-based strategy that leverages both audio and image content to generate captions using resource-constrained devices. The datasets utilized include MSR-VTT and TREC-VTT22. We have developed an application tailored for resource-constrained devices that utilizes the optimal model resulting from our training process. Both modalities of data are then combined and processed by the model to generate a comprehensive description related to the captured data. The primary contribution of this work lies in the introduction of an innovative end-to-end application that leverages audio and image data. This application can be utilized on a mobile device to autonomously produce descriptions.**

I. Introduction

Video captioning involves converting video content into textual descriptions. In video processing, a common approach involves combining frames to generate captions, often utilizing techniques like 3D convolution. This method increases complexity by requiring information extraction from the temporal dimension [1], which can be challenging within the mobile devices context.

Moreover, videos can include audio, which introduces both potential and complexity to this task [1]. These attributes, combined with the limitations imposed by mobile devices, present challenges for autonomously conducting video captioning, i.e., without external resources such as internet connectivity and access to powerful models.

In the realm of mobile or resource-constrained devices, a range of strategies has been employed to surpass these limitations, including distillation, pruning, and quantization [2]. Moreover, numerous deep learning models, proven effective for tasks like classification and segmentation, can perform admirably under such constraints. When addressing video captioning, one can adapt and utilize these models to tackle the various aspects of the task, capitalizing on their high accuracy even within compact architectures and minimal resource consumption.

An example in the context of image captioning is the work by [3], in which the authors use TinyBert [4] in the development of an captioning framework named LightCap that was tailored for resource-limited devices. In a broader context, studies have demonstrated the utility of the audio modality in enhancing caption generation [5]–[8].

Specifically, audio information complements the visual aspect, for instance, when capturing speech or sounds associated with events occurring. Nevertheless, the exploration of this information in the context of video captioning on mobile devices remains limited.

The aim of this study is to evaluate whether integrating image and audio data enhances video captioning performance in scenarios involving resource-constrained devices. To conduct our experiments, we employed an Encoder-Decoder transformer-based architecture and explored three training strategies: a) utilizing solely image features, b) utilizing only audio features, and c) utilizing both feature types. While it is recognized that audio content within videos can enhance video descriptions, this aspect remains relatively unexplored within the realm of mobile devices, where hardware limitations present significant challenges.

The paper is structured as follows. We introduce recent literature related to Video Captioning, Multimodality, and Mobile Devices in Section II. Section III outlines the application framework, experiment setup, and training configurations used in our study. In Section IV, we present the preliminary results obtained from our experiments. Lastly, Section V provides the paper's conclusion and outlines the next steps we intend to take.

II. Related Works

Numerous studies have tackled the challenge of transforming video content into textual descriptions. For example, in the work by [9], a model named VPCSum was introduced to create paragraphs from video data. The approach adopted by the researchers encompasses three key components: 1) image selection; 2) depiction of chosen images, and 3) synthesis of the descriptions. [10] proposed an analytical framework designed for video surveillance, which can identify objects and establish their connections to events.

Related to the usage of both image and audio data, the work [11] introduced a deep neural approach that employs multiple modes to generate detailed captions for videos. The authors developed a model consisting of event detection and two neural networks: a 3D convolutional network for processing sets of frames, and a VGG-based network for audio data.

Regardless these work's importance, they have not focused on mobile devices. In this way, the work [12] created an application responsible for converting video content into audio

descriptions, which was implemented on ARM-based processor hardware. The researchers utilized a series of specialized models for fine-grained object classification, each focusing on a specific category. These objects are then transformed into audio using a text-to-speech library.

In [13], the authors have developed a smart device utilizing ARM-based processor hardware for video surveillance. The system incorporates a YOLO network that receives data captured by a video camera triggered by motion detected through an infrared sensor. The primary objective was to create a system capable of identifying individuals and environmental intrusions.

The work by [14] involved the development of a neural network with residual connection features, which was then integrated into an application named WeCapV2 designed for Android systems. As per the authors, this application autonomously generates video descriptions. The proposed model underwent training using the Microsoft Research Video Description Corpus (MSVD) dataset and was subjected to evaluation using metrics such as BLEU-n, ROUGE-L, SPICE, and CIDEr. The model comprises an "Encoder" featuring CNN and RNN structures, along with a "Decoder" equipped with RNN, embedding, and fully-connected layers. The RNNs utilized are based on multi-layered GRUs, which address the challenge of the "vanishing gradient" problem. During the integration of the model into an Android operating system application, dynamic quantization was applied, and the initially developed PyTorch model was converted into the "TorchScript" format. It is important to note that, in contrast to our work, the training was solely focused on the image domain, given that the MSVD dataset lacks audio data.

In [15], the authors developed the MoviNet network, which deals with optimizations of 3D video networks for mobile devices. In this context, many operations of 3D video networks require a set of frames to be processed at once, which limits the feasibility of these networks on resource-constrained devices. In this work, three strategies were employed to optimize the networks: a) neural architecture search; b) stream buffer; and c) temporal ensemble. These type of network could also be utilized in a different task, similar to their application in our work.

The work [5] applied three audio and image-based approaches to tackle the task of "Video Captioning." The argument put forth is that the auditory channel can provide context in describing the content of videos. The authors developed the following approaches: concatenation of image feature vectors and Mel-Frequency Cepstral Coefficients; weight sharing between the initial layers fed separately with image and sound information; and weight sharing among intermediate and final layers. The authors evaluated these strategies on the MSR-VTT and MSVD datasets, with the latter focusing solely on the visual modality, as the videos lacked audio tracks. The results demonstrated that audio can indeed contribute context to the generation of descriptions for videos.

Differently, our focus lies in strategies aimed specifically at mobile devices, encompassing an end-to-end application responsible for capturing, pre-processing, and generating descriptions in near real-time. Furthermore, our application uses feature extractors tailored for resource-constrained devices.

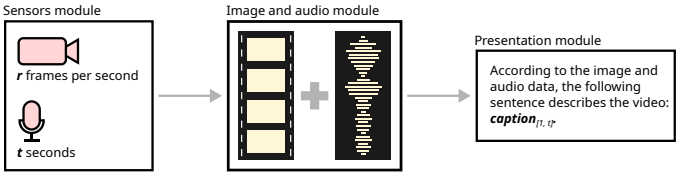


Fig. 1. Application's framework for mobile device video captioning.

III. Methods

A. Application's Framework

Figure 1 presents the application design for performing video captioning with data captured from mobile devices. The framework is composed of three modules: i) sensors module, ii) image and audio module, iii) presentation module. The sensors module refers to the hardware responsible for capturing and storing the data from the image and audio modalities on disk.

A neural network based on Encoder-Decoder constitutes the image and audio module. Initially, a set of images and a portion of audio data are processed through efficient mobile neural networks such as ConvNeX Tiny and YAMNet, with the objective of extracting image and audio features, respectively.

The neural networks used to extract features from the image and audio modalities were versions pre-trained on ImageNet and AudioSet, respectively. In the case of ConvNeXt Tiny, pre-trained on the ImageNet dataset, the input resolution is 224 by 224 pixels. The selected versions are the most compact in terms of size, aligning with the goal of integrating them into the application designed for mobile devices. Our research focuses on training the decoder component, responsible for text generation. This is achieved by utilizing the feature vectors generated by these network backbones as input.

The subsequent phase involves employing these feature vectors for the purpose of feeding the captioning model. Subsequently, this model generates a caption constrained within a predefined vocabulary and a specified word count. The application employs this information to furnish the user with a description of the scene captured by the sensors of the mobile device. The entire process occurs internally and autonomously within the device. This Work in Progress paper encompasses the initial stages of our research, which involve training audio and image-based networks. The subsequent steps will entail amalgamating these two types of networks into a single multimodal network to harness both audio and image data contexts.

B. Experiment Setup

1) Datasets: We have trained the multimodal video captioning model using both the "TREC 2022 Video-to-Text" (TREC-VTT22) dataset and the "Microsoft Research Video to Text" (MSR-VTT) dataset [16]. The TREC-VTT22 dataset

is provided by the "Text Retrieval Conference (TREC)" for a competition and contains a total of 1,000,000 small video segments. For our study, we focused on a subset of approximately 2,000 videos from this dataset, each accompanied by five descriptions[1]. The MSR-VTT dataset comprises 10,000 videos, each with 20 associated captions.

2) Pre-processing: Before commencing the training process, we extracted and stored image and audio file features from the MSR-VTT and VTT22 datasets in Numpy format (*.npy). Concerning images, we took into account 80 frames per video, resulting in matrices with dimensions of $80 \times 7 \times 7 \times 768$. The final three values, $7 \times 7 \times 768$, indicate the output dimensions of the ConvNeXt Tiny-based feature extractor.

Regarding the audio feature extractor, YAMNet generates matrices with dimensions of $(N, 1024)$, where N signifies the count of 0.48-second audio segments in each video. For the VTT22 dataset, $N = 27$, while for MSR-VTT, $N = 74$; these values correspond to the medians of video durations in the VTT22 and MSR-VTT datasets, respectively.

In the case of the captions, all annotations in the dataset underwent preprocessing to remove punctuation, convert characters to lowercase, and eliminate double spaces.

3) Training: We have trained a set of neural networks considering solely the audio modality and solely the image modality. Furthermore, we present the results of the training procedure conducted using the MSR-VTT dataset. The hardware used during the training phase is composed of dual GPU NVIDIA A5000, each with 24GB of RAM. For training and evaluating the models, we have used the TensorFlow framework in a Jupyter Notebook environment.

During the training phase, we experimented with various model configurations, including vocabulary size ranging from 1,500 to 5,000, different caption lengths, number of heads, and number of layers. While getting the best values with respect to caption and vocabulary size, we conduct training of several audio-based and image-based networks varying the number of heads and layers. The utilization of VTT22 will be carried out in the subsequent stages, focusing on training a network incorporating both types of data.

C. Mobile Application

The end-to-end application was implemented on a Raspberry Pi 4 Model B, equipped with 8GB of RAM and a processor based on ARM-v8 architecture. We selected this hardware for its versatility, as it allows the integration of additional components such as batteries, cameras, and sensors. Despite having a large amount RAM, our intention was to utilize only portion of it with a 64-bit Raspberry Pi system loaded Desktop environment. Furthermore, it is possible to adapt the mobile application for use on smartphones or other resource-constrained devices.

[1]We obtained the videos and descriptions from https://trecvid.nist.gov/trecvid.data.html, accessed on March 10, 2023.

We have used the Python programming language in the deployed application which uses the TensorFlow Lite framework. Figure 2a depicts a Raspberry Pi 4 Model B, equipped with a video camera, along with an example of a captured frame. Figure 2b illustrates a typical scenario showcasing the potential applications of the mobile application: a park scene with sidewalks, trees, and shadows, where pedestrians could appear at times.

IV. Results

This section presents the preliminary results we have obtained concerning strategies based solely on audio and solely on image data. All training procedures were standardized with regard to the following parameters: a vocabulary size of 2,500 and a maximum limit of 40 words for the generated descriptions, values that we chose in previous training stages.

In Table I, we display the masked accuracy values across thirteen distinct architecture configurations, varying the number of heads and layers. The metric "masked accuracy" is related to a masked language model, where, during the training phase, random words are replaced by a special token (usually represented by the term "[MASK]"), and then the goal is to predict which word originally existed there before the replacement [17]. In the next stages of this work, we will compute the BLEU-n, CIDEr and other metrics that are related to image and video captioning tasks.

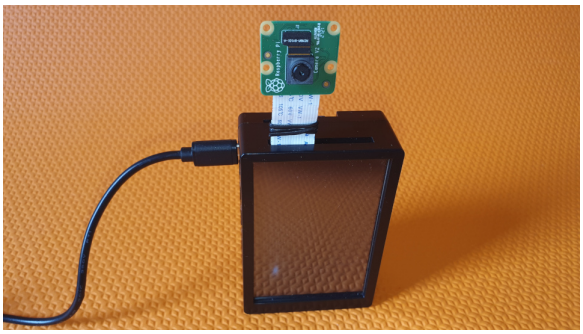
TABLE I
Masked accuracy for the audio-based networks.

Name	Heads	Layers	Parameters	Size (MB)	Masked Acc.
$A1$	10	10	95,846,340	365.62	0.3319
$A2$	10	2	20,203,460	77.07	0.3363
$A3$	2	10	22,323,140	85.16	0.3445
$A4$	4	4	17,057,220	65.07	**0.3547**
$A5$	6	6	35,967,940	137.21	0.3541
$A6$	8	8	62,230,980	237.39	0.3318
$A7$	1	1	2,476,740	9.45	0.3355
$A8$	2	2	5,498,820	20.98	0.3469
$A9$	4	2	9,174,980	35.00	0.3277
$A10$	2	4	9,704,900	37.02	0.3411
$A11$	2	6	13,910,980	53.07	0.3489
$A12$	1	8	10,764,740	41.06	0.3230
$A13$	2	7	16,014,020	61.09	0.3508

Network model $A4$ exhibits the highest masked accuracy while maintaining a small size (approximately 65 MB), although still larger than the most compact architecture, $A7$[2], which has a size of 9.45 MB. We can observe that having more heads and layers does not necessarily equate to higher accuracy; instead, it results in a larger size.

Table II presents several examples of captions generated by this network and the network labeled $I7$, which was trained exclusively on image features. We also display the BLEU-2 and BLEU-3 scores for these predicted captions with respect to the ground-truth data available in the datasets; the BLEU-4 scores resulted in zero.

[2]The "A" in $A7$ stands for the word "Audio." We use this notation to indicate whether a network was trained with only audio ($A7$), images ($I7$), or both images and audio ($IA7$).



(a) Rasberry Pi 4B and video camera

(b) An example of frame captured by the mobile application

Fig. 2. Raspberry Pi 4B equipped with a video camera and an example of captured frame.

TABLE II
EXAMPLES OF CAPTIONS AND METRICS BLEU@2-3 (B@2-3) OBTAINED BY AUDIO AND IMAGE-BASED NETWORKS (N) FOR VIDEOS IN THE MSR-VTT (M) AND VTT22 (V) DATASETS.

N	ID	Caption	B@2	B@3
$A4$	$M9004$	A man is cooking food	0.165	0.140
$I7$		A woman is cooking food	0.233	0.222
$A4$	$M1990$	A man is singing	0.707	0.630
$I7$		A man is talking about a man	0.690	0.000
$A4$	$M3106$	A man is talking about the benefits of the car	0.558	0.427
$I7$		A man is performing a stage	0.437	0.000
$A4$	$V1010$	A man is talking about a man in a man	0.221	0.000
$I7$		A man in a white shirt is walking down a runway	0.248	0.000

On one hand, we can observe that the audio modality effectively captured the cooking action, as expected for video ID $M9004$. We use the notation M to indicate that the video belongs to the MSR-VTT dataset, and the character V corresponds to VTT22. This particular video depicts a woman cooking food, and one of the corresponding ground truth captions reads: "A woman using a red spoon cooks and stirs pieces of meat and onion in a pan."

On the other hand, the image modality managed to identify the person in the video as a woman. However, many objects were not detected or explicitly mentioned in the generated caption. Increasing the vocabulary size can be effective in addressing undetected objects, which will be carried out in the upcoming stages of this research.

Regarding the video $V1010$, the audio modality provides information about an urban scene, which is indicated by the term "runway." The image modality was not able to detect or provide a meaningful caption. It is important to note that network $I7$ has only one layer and one head in its architecture. Additional configurations will undergo training and are anticipated to result in more meaningful captions, akin to those generated by the audio-based networks featuring four layers and four heads.

We intent to take into account strategies tailored for machine learning model deployment on mobile devices, such as knowledge distillation. Throughout this technique, the work by [18] has reduced the inference time by 80%, with a small drop in captioning accuracy.

V. CONCLUSION

In this work, we delve into techniques aimed at enabling video captioning on mobile devices by utilizing pre-existing audio and image feature extractors. We implement an Encoder-Decoder architecture based on transformers, providing the network with features previously extracted from these modalities' data. The first technique employed was solely based on the auditory modality and was capable of detecting significant elements present in the video sequences of the MSR-VTT and VTT22 datasets. The second technique captured the inherent presence within the visual channel.

The combination of both strategies will be the next step to be pursued in this study, along with conducting further experiments involving the image modality. It is worth noting that the models utilized are suitable for deployment on devices with limited computational resources, such as smartphones. The objective is to investigate these strategies within an application for these devices autonomously, which means without reliance on internet resources and more powerful models.

In our initial experiments, the best model generated through our training procedures has been integrated into a mobile application deployed on Raspberry Pi hardware equipped with a camera and microphone. The application records and preprocesses data from both modalities, feeding the network to generate a comprehensive description related to the captured data.

ACKNOWLEDGMENT

The authors would like to thank FAPESP (grants #2015/22308-2, #2022/12204-9, #2022/15304-4), CNPq, CAPES, FINEP and MCTI PPI-SOFTEX (TIC 13 DOU 01245.010222/2022-44) for their financial support for this research.

REFERENCES

[1] M. Abdar, M. Kollati, S. Kuraparthi, F. Pourpanah, D. McDuff, M. Ghavamzadeh, S. Yan, A. Mohamed, A. Khosravi, E. Cambria, and F. Porikli, "A review of deep learning for video captioning," 2023.

[2] Y. Wang, J. Wang, W. Zhang, Y. Zhan, S. Guo, Q. Zheng, and X. Wang, "A survey on deploying mobile deep learning applications: A systemic and technical perspective," *Digital Communications and Networks*, vol. 8, no. 1, pp. 1–17, 2022.

[3] N. Wang, J. Xie, H. Luo, Q. Cheng, J. Wu, M. Jia, and L. Li, "Efficient image captioning for edge devices," 2022.

[4] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," 2020.

[5] W. Hao, Z. Zhang, and H. Guan, "Integrating both visual and audio cues for enhanced video caption," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[6] Y. Tian, C. Guan, J. Goodman, M. Moore, and C. Xu, "An attempt towards interpretable audio-visual video captioning," *arXiv preprint arXiv:1812.02872*, 2018.

[7] V. Iashin and E. Rahtu, "A better use of audio-visual cues: Dense video captioning with bi-modal transformer," *arXiv preprint arXiv:2005.08271*, 2020.

[8] Y. Shen, L. Yang, L. Wen, H. Yu, E. Elhamifar, and H. Wang, "Exploring the role of audio in video captioning," *arXiv preprint arXiv:2306.12559*, 2023.

[9] H. Liu and X. Wan, "Video paragraph captioning as a text summarization task," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021, pp. 55–60.

[10] C. M. Fonseca and J. G. S. Paiva, "A system for visual analysis of objects behavior in surveillance videos," in *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2021, pp. 176–183.

[11] V. Iashin and E. Rahtu, "Multi-modal dense video captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[12] A. Karkar, J. Kunhoth, and S. Al-Maadeed, "A scene-to-speech mobile based application: Multiple trained models approach," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. IEEE, 2020, pp. 490–497.

[13] V. Viswanatha, R. Chandana, and A. Ramachandra, "Iot based smart mirror using raspberry pi 4 and yolo algorithm: A novel framework for interactive display," *Indian Journal of Science and Technology*, vol. 15, no. 39, pp. 2011–2020, 2022.

[14] S. Aydin, Ö. Çayli, V. Kiliç, and O. Aytuğ, "Sequence-to-sequence video captioning with residual connected gated recurrent units," *Avrupa Bilim ve Teknoloji Dergisi*, no. 35, pp. 380–386, 2022.

[15] D. Kondratyuk, L. Yuan, Y. Li, L. Zhang, M. Tan, M. Brown, and B. Gong, "Movinets: Mobile video networks for efficient video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 020–16 030.

[16] 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.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[18] Ö. Çaylı, X. Liu, V. Kılıç, and W. Wang, "Knowledge distillation for efficient audio-visual video captioning," *arXiv preprint arXiv:2306.09947*, 2023.