Diyala Journal of Engineering Sciences

Journal homepage: https://djes.info/index.php/djes

ISSN: 1999-8716 (Print); 2616-6909 (Online)

Real-time Arabic Video Captioning Using CNN and Transformer Networks Based on Parallel Implementation

Adel Jalal Yousif [1,*], Mohammed H. Al-Jammas [2]

[1] Department of Computer Engineering, University of Mosul, Mosul, Iraq
[2] College of Electronics Engineering, Ninevah University، Mosul, Iraq

ARTICLE INFO

Article history:
Received September 21, 2023
Revised February 10, 2024
Accepted February 19, 2024
Available online March 6, 2024

Keywords:
Arabic video captioning
Parallel architecture
Deep learning
Video description
Real-time captioning

ABSTRACT

Video captioning techniques have practical applications in fields like video surveillance and robotic vision, particularly in real-time scenarios. However, most of the current approaches still exhibit certain limitations when applied to live video, and research has predominantly focused on English language captioning. In this paper, a novel approach for live real-time Arabic video captioning using deep neural networks with a parallel architecture implementation is introduced. The proposed model primarily relied on the encoder-decoder architecture trained end-to-end on Arabic text. Video Swin Transformer and deep convolutional network are employed for video understanding, while the standard Transformer architecture is utilized for both video feature encoding and caption decoding. Results from experiments conducted on the translated MSVD and MSR-VTT datasets demonstrate that utilizing an end-to-end Arabic model yielded better performance than methods involving the translation of generated English captions to Arabic. Our approach demonstrates notable advancements over compared methods, yielding a CIDEr score of 78.3 and 36.3 for the MSVD and MSRVTT datasets, respectively. In the context of inference speed, our model achieved a latency of approximately 95 ms using an RTX 3090 GPU for a temporal video segment with 16 frames captured online from a camera device.

1. Introduction

These days, multimedia content (e.g., images, text, videos, and audio) can be easily generated and shared over the internet. Due to the growing popularity of mobile devices and the concurrent growth in storage capacity and internet bandwidth, video has become a dominant and widely consumed form of multimedia content. The need for advanced video understanding techniques has motivated researchers to improve and enhance these capabilities by developing various machine learning models and innovative algorithms. Video captioning aims to describe the visual information of a video clip by generating a natural language sentence according to the video contents [1]. Hence, video captioning plays a significant role in bridging the gap between the fields of computer vision and natural language processing. This integration of vision and language enables various real-life applications such as video surveillance [2], human-robot interaction [3], assistance for visually impaired people [4], and many others.

Video captioning is recognized as one of the most challenging tasks that needs a high-level understanding of video content. This involves identifying salient objects, activities, and relations, followed by the generation of a

* Corresponding author.
E-mail address: adil.jalal.yousif@uodiyala.edu.iq
DOI: 10.24237/djes.xxxx.13301

meaningful and accurate description for the video. Recently, deep learning encoder-decoder architectures [5] have been extensively used in video captioning methods, providing an effective framework for processing and understanding video content. Within these approaches, Convolutional Neural Networks (CNNs) [6] are often utilized to extract and encode video features into sequences of vectors. Captions are then generated by decoding these vectors with Recurrent Neural Networks (RNNs) [7] or Transformers architecture [8]. Many previous works on video captioning utilized 3D-CNN and optical flow for temporal or motion feature extraction, which proved to be time-consuming and incompatible with real-time processing. On the other hand, Video Swin [9] has demonstrated state-of-the-art performance in video recognition across various datasets [10].

Most of the research in video captioning predominantly focuses on the English language, with some attention given to Hindi, Chinese, and other languages [11]. However, there is currently a lack of studies on Arabic video captioning, prompting our focus on this area. This limitation can be attributed to the absence of publicly available Arabic datasets for video captioning. Moreover, the Arabic language presents challenges due to its morphological complexity and the presence of diacritics in various forms. The AraBERT model [12] is utilized to alleviate some aspects of Arabic morphology. However, to design an efficient video captioning system that meets real-life applications, it is crucial to consider the following key factors:

- Ensuring accurate descriptions
- Addressing the target language
- Minimizing the computational time needed for caption generation

To address the above issues, a novel real-time video captioning method is proposed for describing streaming video in Arabic utilizing a combination of CNN and Transformer networks with a parallel architecture implementation. The main contributions of this work are:

- Designing an end-to-end Arabic video captioning system utilizing deep convolutional networks and transformer architectures.
- Developing a strategy for real-time captioning of live camera streams based on parallel architecture.
- Constructing an Arabic version of MSVD[1] [13] and MSR-VTT[2] [14] datasets using Google Translate API for model training and benchmarking.
- Conducting extensive experiments and comparisons with various deep neural models.

2. Related work

Existing video captioning techniques can be categorized into traditional techniques and modern video captioning techniques. Early works focus on the traditional ones, which are primarily based on template-based methods, as observed in prior studies [15, 16]. In template-based methods, predefined templates or structures are utilized for generating sentences Based on the extracted visual features and semantic concepts. Typically, the extracted semantic concepts include subjects, objects, and verbs based on a set of visual classifiers. Then, it generates descriptions by using a linguistic model that allocates the anticipated triplets to predefined sentence structures. For example, assuming video frames show a person running in a street, the semantic concepts are "person," "running," and "street". Then, if the sentence template is like "[subject] is [verb] in the [object]", the generated caption will be "The person is running in the street".

However, methods based on templates have certain limitations, notably a lack of flexibility due to their dependence on predefined sentence structures. This issue restricts their capability to adapt to a diverse range of video content. Furthermore, the templates need to be carefully designed and adapted to specific datasets, and thus, these methods have limited ability when it comes to handling unseen data.

On the other hand, to overcome these limitations associated with the traditional techniques, neural networks, and specifically

1: https://www.cs.utexas.edu/users/ml/clamp/videoDescription/
2: https://www.microsoft.com/en-us/research/publication/msr-vtt-a-large-video-description-dataset-for-bridging-video-and-language/

deep learning are utilized in the modern techniques which are known as deep learning video captioning methods. In general, the predominant architecture used in deep learning based video captioning methods is the encoder-decoder framework. B. Wang et al. [17] introduced an encoder-decoder reconstruction framework with a dual learning strategy, where CNN is employed as an encoder and LSTM with a temporal attention mechanism is exploited for the decoder part. However, the model's ability to capture spatiotemporal features (motion) was limited, leading to less accurate captions.

H. Ye et al. [18] adopted a hierarchical modular deep neural network for video description. This model utilizes a three-level hierarchy to connect video representations with linguistic semantics for caption generation. Nevertheless, the sequentially three-level hierarchy significantly increased the model complexity due to each level was implemented as a separate component.

A. Singh et al. [11] proposed a video captioning method for the Hindi language using a hybrid attention mechanism. The visual features are extracted from video clips using 3D-CNN, while LSTM with the attention mechanism is used for caption generation. However, the assessment of this work was limited to a single benchmark. Additionally, using only 3D-CNNs restricts comprehensive visual understanding. J. Zhang et al [19] proposed an object-aware video captioning method based on the graph technique. This method utilizes spatiotemporal graphs to represent objects along with their relationships. However, due to the complexity involved in designing the graph structure, the caption generation process required a considerable amount of time, taking 7.73 seconds on the GTX 1080 GPU. S. Zaoad et al. [20] adopted a deep architecture captioning method for the Bengali language. The authors explored several CNN-RNN frameworks with an attention layer to present the best combination of CNNs and RNNs. However, the evaluation of this method was restricted to a single benchmark, and motion features were ignored. S. Ma et al. [21] adopted a deep neural network live commenting method to generate short and quick Chinese comments for videos. The video is sampled at 1fps and for describing the video at a certain time-stamp. However, the generated comment is short and very simple. Furthermore, the computational time was not investigated. Y. Chen et al. [22] adopted the first study that explored online streaming video captioning. The video frames are sampled at 1fps and sequentially fed the sampled frames to PickNet, to decide whether to pick the current frame or discard it. Then, a pre-trained 2D-CNN extracts its visual features. The caption is generated by applying an image captioning method for the selected frame. However, handling spatiotemporal features was limited. Moreover, hardware specification and time-consuming analysis were not reported

3. Methodology of proposed model

In this section, our live real-time Arabic video captioning approach is presented including video representation, caption generation, and streaming video processing. The architecture of the proposed video captioning method is shown in Figure 1.

3.1 Visual encoder

The videos in both the MSVD and MSR-VTT datasets contain 30 frames per second with variable time durations. This high frame rate often leads to redundant information, especially in consecutive frames. Neural network models require input data with a fixed shape for all samples. Hence, to eliminate redundancy and decrease computation time, all videos are uniformly sampled to N equally spaced frames ($N = 16$ is chosen).

To generate captions that accurately capture both the static scene (i.e. spatial features) and dynamic events scene (i.e. motion or temporal features), our approach utilizes two visual representation models. The pre-trained Inception-v4 convolutional network is employed in this work for spatial feature extraction, producing a 1536-feature vector for each processed frame. Inception-v4 is notably recognized for achieving a favorable balance between speed and accuracy. The architecture of Inception-v4 is characterized by multiple

parallel convolutional layers and reduction blocks, which effectively reduce spatial dimensions while maintaining performance. Simultaneously, Video Swin, which represents a state-of-the-art video transformer model, is employed to analyze the relationships and motion between a sequence of frames. The Video Swin-B implementation is utilized for better accuracy-speed trade-off. This model was trained on Kinetics-600 and ImageNet-21K datasets, which include 600 action classes and 21,000 appearance classes respectively. The last classification layer is replaced with a global average pooling layer to convert the resulting features from (1024 x 8 x 7 x 7) into (1024 x 1 x 1 x 1) while retaining vital information. Hence, the shape of the extracted features by Inception-v4 and Video Swin-B for each sampled video is Nx1536 and Nx1024 respectively. Instead of simple feature concatenation, a neural feature fusion strategy is adopted in our approach as shown in Figure 2.

This strategic fusion not only creates a more compact and informative representation of the video content but also significantly reduces computational time, essential for real-time captioning. Finally, the resulting feature is passed to the transformer encoder to apply the self-attention mechanism to ensure that the model can effectively capture and emphasize relevant information within the feature representation.

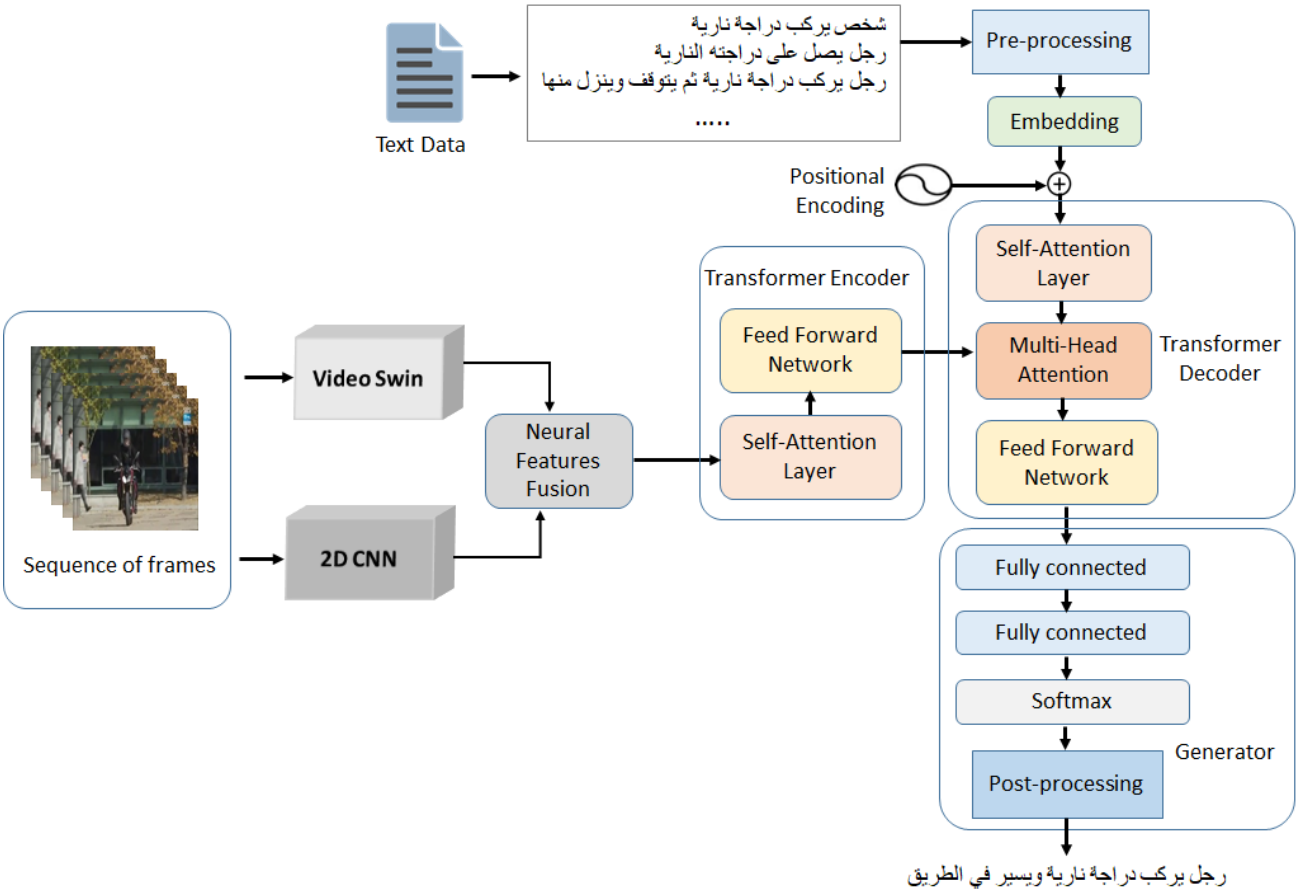

Figure 1. Structure of the proposed video captioning method

3.2 Caption decoder

Our caption decoder consists of two parts: decoder and generator. The decoder, which utilizes the Transformer decoder architecture, receives the video representation features and the word embedding vectors as input and produces caption features. The generator predicts the next word in the caption by calculating the probability of each word over the constructed vocabulary using softmax activation, based on the decoder output.

Subsequently, the word with the highest probability is selected as the predicted word.

The caption decoder progressively anticipates the next word, starting from an initial token <SOS>. In each step of the iteration, it receives the video representation features and a partial caption that has already been predicted. Then predicts the next word by passing through the decoder and subsequently the generator, where each word is assumed to be transformed into a word embedding vector. The generator uses a greedy search in selecting the next word, choosing the word with the highest probability until a distinctive ending token <EOS> is encountered. Finally, the post-processing returns the segmented words to their original form by using the AraBERT as a de-segmenter.

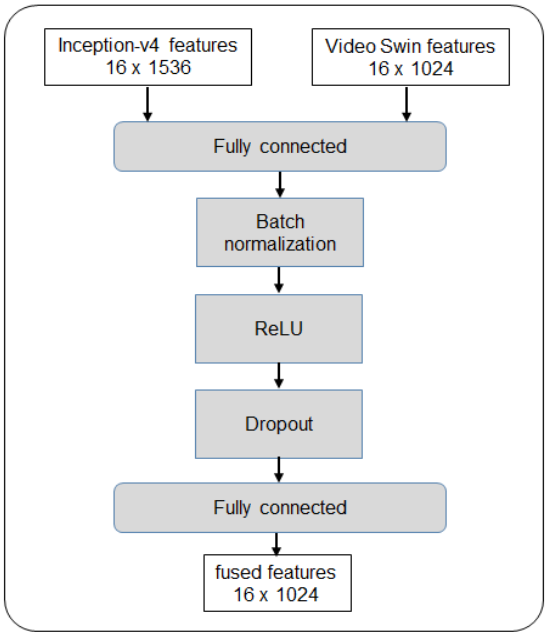

Figure 2. Trainable neural network for feature fusion

3.3 Live streaming video processing

Most existing approaches of video captioning operate in offline mode, assuming the entire video is available at the start of processing. However, real-life applications often involve streaming video inputs where the current scene needs to be continuously updated. However, handling videos as streams introduces some challenges, such as limited window size can significantly reduce long-term context capturing or lead to significant processing delays. To address this issue, an efficient strategy is adopted for live streaming video processing captured from a camera device, while preserving our encoder-decoder architecture without modification.

Our strategy involves partitioning the incoming video frames into segments of N frames, aligning with the number of frames processed by the network. A working memory W_m is employed to consistently involve N sampled frames fed to the captioning model with a timestamp. For filling the working memory, while the incoming sequence of frames is played at 30 frames per second, one frame is uniformly sampled every 6 consecutive frames (i.e. 5 fps), maintaining a continuous update of frames. When the selected frames accumulate to N, they are fed into our captioning model for real-time caption generation. Then, with each new time segment, additional N frames join the processing stream, causing the older half of the samples in the working memory to be swapped with samples from the new segment. This dynamic process ensures that the model can predict real-time changes while maintaining long-term context understanding through the utilization of the working memory.

Furthermore, in certain real-life scenarios, when there is no activity in front of the camera to describe, a movement detection scheme is implemented before the caption generation process. This step is implemented to shift our captioning model into an idle state, ensuring the avoidance of unnecessary consumption of computational resources and improving its compatibility with various real-world applications. For every 6 consecutive frames, the similarity index (SIM) is calculated between the first and last frames after converting them into grayscale. Then, a movement is identified if the similarity index is less than a predefined threshold. The overall live streaming video processing steps are presented in Algorithm 1. Additionally, an overview of the streaming video processing scheme is depicted in Figure 3.

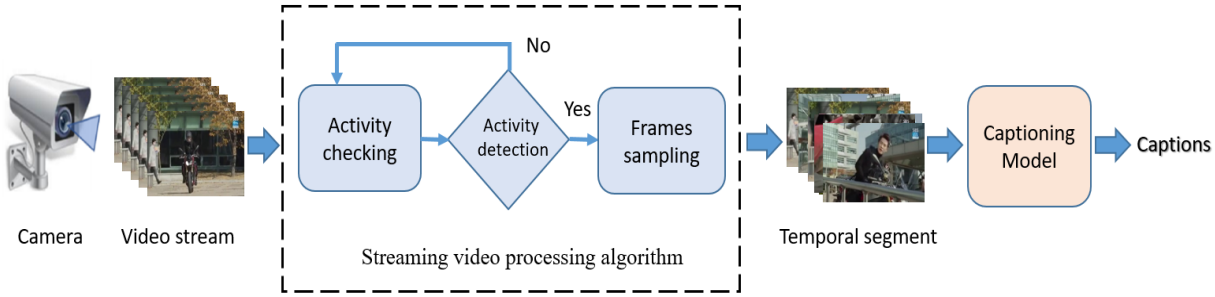

Figure 3. Overview of the streaming video processing scheme

Algorithm 1: Streaming video processing

Input: live streaming video (V), Number of Samples (N), threshold (Th)
Output: segment of N frames

Create an empty queue Q to hold incoming frames;
Prepare working memory W_m ;
while new frames f_{i+t} are coming from V **do**
  Append frame f_i to the end of queue Q
  if i % 6 == 0 **then**
    Calculate SIM between f_i and f_{i-5}
    if SIM > Threshold **then**
      Empty W_m
      Continue
    else: do
      Add frame f_i to W_m
      if W_m is filled with N frames **then**
        Output the segment of N frames to feed to the captioning model
        Discard the older $N/2$ frames from W_m
      end if
    end if
  end if
end while

4. Experiments

4.1 Datasets and evaluation metrics

In this section, the proposed video captioning method is assessed using the most widely used benchmarks which are the Microsoft Research Video Description Corpus (MSVD) [13] and the Microsoft Research Video to Text (MSR-VTT) [14]. The MSVD combines 1970 videos with 40 captioning sentences on average. The broadly used data splitting for the MSVD dataset is adopted: 1200, 100, and 670 clips for training, validation, and testing respectively. MSR-VTT is a frequently used benchmark dataset in video captioning, presenting a more challenging task compared to MSVD.

It consists of 10,000 videos with 20 captioning sentences for each. The standard data splitting for the MSR-VTT dataset: 6513, 497, and 2990 clips for training, validation, and testing respectively. For performance evaluation and comparison, the most frequently used evaluation metrics in video captioning are investigated including Bilingual Evaluation Understudy (BLUE) [23], Recall Oriented Understudy for Gisting Evaluation (ROUGE) [24], Metric for Evaluation of Translation with Explicit Ordering (METEOR) [25], and Consensus-based Image Description Evaluation (CIDEr) [26]. These metrics assess the quality of the predicted captions by measuring the similarity between the generated and the human-annotated captions.

4.2 Implementation details

To elevate some of the morphological complexity of Arabic text, the deep neural AraBERT model is utilized as a word segmenter. The AraBERT model separates most of the prefixes and suffixes that are appended to words such as the definite article, conjunctions, and pronouns. For more details about AraBERT, please refer to [12]. Moreover, incorporating the AraBERT model improves the overall performance of our captioning model by reducing the vocabulary size by about half by eliminating redundant words. For both MSVD and MSR-VTT, captions longer than 20 tokens

are trimmed, and shorter ones are padded with a special token. our vocabulary is constructed by choosing the unique words or tokens present in the dataset after removing the rare words and reducing redundancy utilizing the AraBERT model. Consequently, the vocabulary size, representing the number of unique tokens, is 4358 tokens for MSVD and 6112 tokens for MSR-VVT.

For the encoder side, each video is sampled into 16 equally-spaced frames (i.e. N = 16 frames) and fed them to Inception-v4 and Video Swin-B models for feature extraction. The shape of tensor features of Inception-v4 is 16x1536, while Video Swin-B produces a spatiotemporal feature vector with a shape of 1x1024, which is repeated 16 times to be compatible with the Inception-v4 features. After passing both features together to a neural features fusion network, the resulting feature shape will be 16x1024 rather than 16x2560. Both fully connected layers in this network have 1024 neurons as hidden dimensions. The dropout ratio is set to 0.2. Additionally, batch normalization and dropout are used as regularization methods to mitigate overfitting and facilitate faster convergence in our training process. The primary aim of this network is to minimize the overall model complexity and computational time.

In the Transformer encoder module, a single encoder layer is adopted with a single attention head in the self-attention layer. Both the model dimension and hidden dimension of the feed-forward network are 1024. The feed-forward network consists of two fully connected layers, with Rectified Linear Unit (ReLU) activation applied following the first layer. As a result, both the input and output of the Transformer encoder module have a shape of (16x1024).

In the Transformer decoder module, the same configuration (number of layers and hidden dimensions) of the Transformer encoder is adopted. Additionally, word embeddings are implemented by trainable embedding layers with a dimension of 1024. Hence, the output of the Transformer decoder module also has a shape of (16x1024). For the generator module, the hidden size of both fully connected layers is equal to the vocabulary size, which is 4358 for MSVD and 6112 for MSR-VVT. The first fully connected layer is followed by a ReLU activation function and dropout with a probability of 0.2. The objective of the generator module is to map the feature vectors received from the decoder to the probability distribution of each word in the constructed vocabulary. As a result, the input dimension of the generator is (16x1024), while the output has a shape of (1 x vocabulary size).

For training details, the Adam optimizer is adopted with the following settings: the learning rate is set to 2e-5 and 5e-5, the batch size is set to 128 and 64, and training epochs are set to 30 and 20 for MSVD and MSR-VTT, respectively. In streaming video processing, the threshold value of the similarity index is set to 0.95. The model's parameters are carefully chosen through extensive experimentation, exploring various values to achieve optimal performance trade-offs in terms of speed and accuracy.

4.3 Results and comparison

To our knowledge, there has been no reported work on Arabic video captioning to date. Hence, comparing the system's performance with other models is a challenging task for Arabic captioning as well. To tackle this issue, two existing works for English video captioning are implemented, which are denoted as RecNet [17] and HMN [18]. These model are implemented based on the same configurations provided on their GitHub links. Subsequently, Google Translate is utilized to translate the actual captions predicted by these works into Arabic. This is done for generating Arabic captions, which will be used later for comparison with our model. Furthermore, an Arabic adaptation of RecNet (denoted as RecNet_Ar) is created by training the RecNet model with our Arabic version of MSR-VTT and MSVD in an end-to-end manner. This involves incorporating necessary modifications for Arabic text preprocessing. Therefore, these three methods are employed as baseline models for the purpose of evaluating and comparing our model.The performance evaluation of our method in comparison to the baseline models is illustrated in Table 1. For qualitative evaluation, Figure 4 presents examples of the predicted

captions of our method for the MSR-VTT benchmark. According to the findings in Table 1, our model demonstrates superior performance compared to the baseline models for both datasets, particularly in the CIDEr metric. It is important to highlight that CIDEr is designed specifically for the evaluation of captioning tasks. As depicted in Figure 4, our proposed model can generate high-quality captions with more detailed content.

Table 1: Comparative results with previous studies on the MSVD and MSRVTT benchmarks

Dataset	Methods	BLUE4	METEOR	ROUGE	CIDEr
MSVD	RecNet [17]	20.8	36.0	47.7	40.5
	HMN [18]	30.0	39.7	52.9	50.8
	RecNet_Ar	41.7	36.7	50.4	60.0
	Proposed	43.5	39.8	58.1	78.3
MSR-VTT	RecNet [17]	19.5	28.5	40.5	24.7
	HMN [18]	24.1	33.1	47.7	33.0
	RecNet_Ar	25.8	32.1	46.7	35.4
	Proposed	24.5	33.2	47.5	36.3

Figure 4. Qualitative results on the test set of MSR-VTT benchmark.

Table 2 presents a time-consuming analysis of the proposed method at the inference stage. Our model is tested on two different hardware setups using the PyTorch framework: a workstation equipped with an NVIDIA RTX 3090 GPU and a laptop equipped with an NVIDIA RTX 3060 GPU mobile for addressing portability issues. Referring to Table 2, our model is capable of achieving real-time video captioning with an average inference speed of 95 and 300 milliseconds for RTX 3090 and 3060, respectively. The results in Table 2 are obtained for captioning 16 frames captured online from a camera device directly connected to our model. The improved performance of our methods in terms of computational time and caption quality can be primarily attributed to several key factors. Firstly, the utilization of both appearance and motion features extracted by efficient deep neural models: Inception-v4 and the state-of-the-art video Swin Transformer model. This allows our model to capture rich visual information from the input videos in addition to enhancing the quality of the generated captions. Secondly, instead of using the traditional RNNs, the standard Transformer model is utilized for caption generation, which also achieves the best performance in numerous natural language processing tasks due to its ability to capture long-range dependencies and model contextual information effectively. Thirdly, our model is trained end-to-end with

Arabic text rather than using an English-based model and translating the generated captions to Arabic, as in the baseline model. So, the performance of the RecNet_Ar model significantly outperforms that of the original translated RecNet model [17].

Table 2: Inference speed of the proposed system for the encoding-decoding process

Number of sampled frames	CPU core i7-11800H	GPU RTX 3060 Mobile	GPU RTX 3090
16 frames	≈ 3.5 sec	≈ 300 ms	≈ 95 ms

5. Conclusions

In this work, a real-time deep neural model is proposed for describing video content in the Arabic language, which connects video representations with linguistic semantics based on the encoder-decoder framework. The encoder integrates the video features extracted from Inception-v4 and Video Swin utilizing a neural fusion method, while the decoder Transformer is employed for caption decoding. The experiments on the Arabic version of MSVD and MSR-VTT benchmarks demonstrate the effectiveness of our captioning model compared to other existing methods. This research marks a promising advancement in enhancing the accessibility of visual content in native languages within the Arab world, making it suitable for various real-life applications. The proposed model achieved real-time captioning with a latency of approximately 95 ms using a single NVIDIA RTX 3090 GPU. Pros of our research include the provision of an end-to-end Arabic model, providing improved performance, efficient inference speed, and enhanced accessibility for Arabic users. However, it is important to acknowledge some limitations of this work such as the utilization of translated datasets and dependence on hardware.

References

[1] A. J. Yousif and M. H. Al-Jammas, “Exploring deep learning approaches for video captioning: A comprehensive review,” e-Prime - Adv. Electr. Eng. Electron. Energy, vol. 6, no. October, p. 100372, 2023, doi: 10.1016/j.prime.2023.100372.

[2] V. Chundi, J. Bammidi, A. Pegallapati, Y. Parnandi, A. Reddithala and S. K. Moru, "Intelligent Video Surveillance Systems," 2021 International Carnahan Conference on Security Technology (ICCST), Hatfield, United Kingdom, 2021, pp. 1-5, doi: 10.1109/ICCST49569.2021.9717400.

[3] B. Irfan, A. Ramachandran, S. Spaulding, D. F. Glas, I. Leite and K. L. Koay, "Personalization in Long-Term Human-Robot Interaction," 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Daegu, Korea (South), 2019, pp. 685-686, doi: 10.1109/HRI.2019.8673076.

[4] J. O. Williams, “Narrow-band analyzer,” Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[5] A. Khan, A. Khan and M. Waleed, "Wearable Navigation Assistance System for the Blind and Visually Impaired," 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhier, Bahrain, 2018, pp. 1-6, doi: 10.1109/3ICT.2018.8855778.

[6] Adel Jalal, Yousif. "Convolution Neural Network Based Method for Biometric Recognition." Central Asian Journal of Theoretical and Applied Sciences 4.8 (2023): 58-68.

[7] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi and M. Ghogho, "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks," 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, Canada, 2018, pp. 202-206, doi: 10.1109/NETSOFT.2018.8460090.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, Advance Neural Inf. Process. Syst. 30 (2017).

[9] Liu, Ze, et al. "Video swin transformer." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[10] X. Chen, M. Zhao, F. Shi, M. Zhang, Y. He and S. Chen, "Enhancing Ocean Scene Video Captioning with Multimodal Pre-Training and Video-Swin-Transformer," IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society, Singapore, Singapore, 2023, pp. 1-6, doi: 10.1109/IECON51785.2023.10312358.

[11] A. Singh, T. D. Singh, and S. Bandyopadhyay, "Attention based video captioning framework for Hindi," Multimed. Syst., vol. 28, no. 1, pp. 195–207, 2022, doi: 10.1007/s00530-021-00816-3.

[12] Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT, "Transformer-based model for Arabic language understanding", arXiv preprint arXiv:2003.00104, 2020.

[13] D. Chen and W. Dolan, "Collecting highly parallel data for paraphrase evaluation". In ACL: Human Language Technologies- Volume 1. ACL, 190-200, 2011.

[14] 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.

[15] Kojima, A., Tamura, T., & Fukunaga, K., "Natural language description of human activities from video images based on concept hierarchy of actions", International Journal of Computer Vision, 50, 171–184, 2002.

[16] Hanckmann P, Schutte K, Burghouts GJ., "Automated textual descriptions for a wide range of video events with 48 human actions", In: IEEE ECCV, 2012.

[17] B. Wang, L. Ma, W. Zhang and W. Liu, "Reconstruction Network for Video Captioning," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7622-7631, doi: 10.1109/CVPR.2018.00795.

[18] H. Ye, G. Li, Y. Qi, S. Wang, Q. Huang and M. -H. Yang, "Hierarchical Modular Network for Video Captioning," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 17918-17927, doi: 10.1109/CVPR52688.2022.01741.

[19] J. Zhang and Y. Peng, "Video Captioning with Object-Aware Spatio-Temporal Correlation and Aggregation," IEEE Trans. Image Process., vol. 29, no. c, pp. 6209–6222, 2020, doi: 10.1109/TIP.2020.2988435.

[20] M. S. Zaoad, M. M. R. Mannan, A. B. Mandol, M. Rahman, M. A. Islam, and M. M. Rahman, "An attention-based hybrid deep learning approach for bengali video captioning," J. King Saud Univ. - Comput. Inf. Sci., vol. 35, no. 1, pp. 257–269, 2023, doi: 10.1016/j.jksuci.2022.11.015.

[21] S. Ma, L. Cui, D. Dai, F. Wei, and X. Sun, "LiveBot: Generating live video comments based on visual and textual contexts," 33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019, pp. 6810–6817, 2019, doi: 10.1609/aaai.v33i01.33016810.

[22] Y. Chen, S. Wang, W. Zhang, and Q. Huang, "Less is more: Picking informative frames for video captioning," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11217 LNCS, pp. 367–384, 2018, doi: 10.1007/978-3-030-01261-8_22.

[23] Papineni K, Roukos S, Ward T, Zhu WJ. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics, 2002;311–318.

[24] Banerjee S, Lavie A. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, 2005;65–72.

[25] Lin CY. Rouge: A packagefor automatic evaluation of summaries. In: Proceedings of Workshop on Text Summarization Branches Out, Post2Conference Workshop of ACL 2004.

[26] Vedantam R, Lawrence Zitnick C, Parikh D. Cider: Consensus based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015;4566–4575.