Encoder-Decoder Model for Automatic Video Captioning Using Yolo Algorithm

Hanan Nasser Alkalouti
Faculty of Computing and Information Technology
King Abdul-Aziz University
Jeddah, Saudi Arabia
halkalouti@stu.kau.sa.edu

Dr. Mayada Ahmed AL_Masre
Faculty of Computing and Information Technology
King Abdul-Aziz University
Jeddah, Saudi Arabia
Malmasre@kau.edu.sa

2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) | 978-1-6654-4067-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/IEMTRONICS52119.2021.9422600

Abstract— **Humans can use informed visual perception to generate sentences by bridging the gap between the recognition of visual features (images) and linguistic expression (words) describing these images. Videos are an example of visual perception; humans can describe the content of the video in meaningful sentences based on understanding their contents as a caption for the video. However, automating the video caption process is a challenging task as it confronts the model with two problems are: object detection and generating a sentence.**

This research aims to develop a model that automates video captioning based on Encoder-Decoder using a deep learning algorithm following these two steps. Firstly, using the KATNA model to select the most significant frames from the video and remove redundant ones. Secondly, combining the two deep learning algorithms YOLO and LSTM. The You Only Look Once (YOLO) algorithm recognizes objects in the video frames and the Long Short-Term Memory (LSTM) algorithm generates the video caption.

The proposed model describes the video's content in a meaningful sentence and it shows good accuracy and efficiency, it applies YOLO on the MSVD dataset unlike other video captions using other deep learning techniques.

Keywords—(Deep Learning, Natural Language Processing (NLP), Video captioning, You Only Look Once (YOLO)).

I. Introduction

The information over the internet is growing exponentially hour by hour, and visual content understanding becomes an interesting search area in computer vision, and video captioning is one of its applications. There are a huge number of videos uploaded around the world from different areas at one second through the Internet; these videos need to be arranged and classify based on their captions to facilitate reaching them. It is a challenging task to automate the video caption process; tasks related to video captioning are still considered a challenging research topic. The way video stream is structured and how they are dependent on temporal sequencing, multiple frames, varied objects, and actions and generating accurate sentence complicate the captioning process. Deep learning has currently shown up its efficiency in different areas and it has currently transformed computer vision studies and applications, video caption is one of its applications. Research experimenting with video caption using deep learning techniques develop various language models using LSTM (Long short-term memory), RNN (recurrent neural network), CNN (convolutional neural network), GRU (Gated Recurrent Unit), and TPGN (Tensor Product Generation Network).Recently, deep learning showed up its capability to deal with visual and text contents. Based on that, we propose a deep learning model constructs of encoder-decoder architecture; we compare the performance and the accuracy of using YOLO on the MSVD dataset and other deep learning techniques on the MSVD dataset. Our main objective is to develop an Encoder-Decoder video caption that utilizes YOLO as an encoder, unlike other models that use other deep learning techniques. , and compare final results with previous models using other deep learning techniques.

II. Literature Reviews

Video captioning is a process for describing video content using one or more sentences [1]; It translates visual contents to natural language explanation. It starts by recognizing objects and relates them in sentences [2]. Figure1 shows the general process of video captions using Encoder-Decoder. An overview is given in what follows:

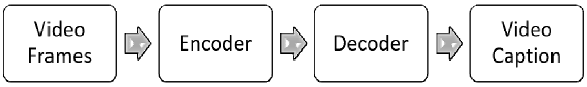


Figure 1. General Structure of Encoder-Decoder Video Caption.

Researchers informed by DL techniques; currently, adapt RNN models, such as LSTM[3] and GRU[4], to act as a decoder of the video clip and learned to generate natural language sentences instead of using a specified template.

In [5], the researchers proposed a RESNET-50 CNN-LSTM Encoder-Decoder for video captioning and an LSTM Encoder-Decoder for sentence generation. There are a different number of video frames; they take samples of each 10 frames to reach an average of 40 processed video frames. In the beginning, the researchers changed the layers' structure of the (CNN) by using the residual function F(x) to enhance performance, and they used it to produce feature vectors of each video frame. A stacked LSTM used for encoding the visual features, and another for decoding the feature output of the CNN to natural language. They used LSTM on both sides of the model as Encoder and decoder; unlike our model, we use it as Decoder and YOLO as Encoder to enhance performance. Besides, it processes a huge number of frames, which consumes time.

In [3], the researcher proposes to enhance the accuracy of video captioning by including Temporal Deformable Convolutional in both Encoder-Decoder. The Temporal Deformable Convolutional (TDConvED) supports combining information of features for a long time by adding fully convolutional to each Encoder-Decoder. In the Encoder, the CNN extracts a feature of video frames to be

978-1-6654-4067-7/21/$31.00 ©2021 IEEE

fed into (TDConvED), model uniformly samples 25 frames of each video. This results in video intervals within context. Mean pooling is used to represent these contexts and send them to the decoder. In the Decoder, stacked shifted convolutional blocks are used to produce a word for each representation. It uses a temporal attention mechanism to help the decoder focus on selected frames based on their weights to produce video captions. It is feed-forward, which means the result from the current layer does not depend on results from previous layers; it affects the accuracy of the final result (caption) in the opposite of using RNN techniques such as LSTM in our model which is back-forward.

In[6], researchers propose a Multimodal Memory Model (M3) for video captions, they proposed a shared memory for both visual (frames) and textual (sentences) and they guide visual attention on described elements to solve visual-textual alignments. The researchers experimented with two datasets: MSVD and MSR-VTT, they uniformly sample 98 video frames to 28, and 149 video frames to 40. Findings demonstrated that their method, when evaluated using BLEU and METEOR, did outperform most of the previously reported methods. It takes a huge number of frames which consume time and there are no specific criteria to choose process frames.

In [2], the Model generates captions based on spatial-temporal attention (STAT); which focuses on some important frames and regions within the video, not whole video frames. Firstly, the encoder network extracts global feature (frame level) using 2d CNN, motion features (frame level) using 3D CNN, and local features (object level (actions)) using faster RCNN from each video frame. Then, the spatial attention mechanism detects the most relevant objects in video frames based on increasing attention weights (sum of global, local, and motion features). After that, the temporal mechanism tracks trajectories of objects detected by spatial and frames; they select frames and regions and send them to the decoder. Finally, LSTM generates sentences with high probabilities of words using beam search. It uses MSVD and MSRVTT-10 datasets, and models evaluated by BLEU, METEOR, and CIDEr.They used RNN techniques on both sides of a model as Encoder and decoder; unlike our model, we use it as a Decoder and YOLO as Encoder to enhance performance.

In [7], they were the first to pick informative frames to be processed by Encoder-Decoder based on Pick Net (two-layer feed-forward neural network) mechanism. Firstly, it transforms each color frame to grayscale and resizes to a small size to produce a "glance" version of frames. Then, subtract the current glance from the previous one (the first frame took to compare with it), results in a flat to fixed-size vector to produce binomial distribution to decide to drop it or keep it. Secondly, kept frames are access to CNN encoder to extract features from them. After that, they use a gated Recurrent Unit (GRU) decoder to generate sentences; effective and performance of generating caption is affected by several selected frames. The model uses Microsoft Video Description (MSVD) and the MSR Video-to-Text (MSRVTT) datasets and researchers evaluated using BLEU, ROUGE, METEOR, and CIDEr. It reduces processing time; it takes between 6 to 8 frames, but other factors should be considered in selecting frames for accuracy.

Many types of research applied deep learning to the MSVD dataset; but they did not apply with YOLO, they use other deep learning techniques as mentioned. Therefore, the researchers in this paper compare their results with the result of the proposed model that YOLO applies to MSVD.

III. METHODOLOGY

The proposed model has four main elements: dataset, keyframe extraction, object detection, and sentence generation process. These elements join a sequential process to generate a video caption (Figure 2).

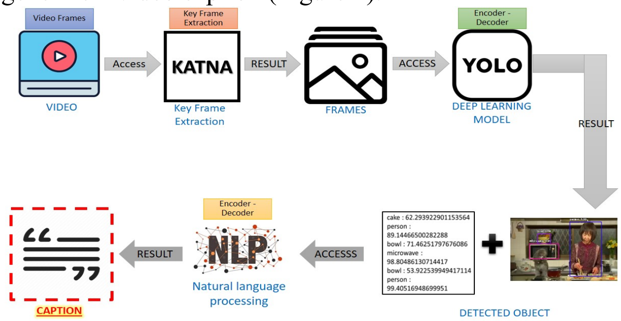


Figure 2: Architecture of Proposed Model.

For the dataset, we choose MSVD; it contains 1970 open muted videos collected from YouTube. The average duration of each video is 5 -25 seconds and it is about one action. There are 41 captions for each video on average. Researchers mostly use it for video captioning, and it is divided into training, testing, and validation sets[9]. Video caption based MSVD is done through the next three processes:

A. Key Frames Extraction (using KATNA):

keyframe extraction is a technical process for capturing meaningful frames from videos. Videos contain heavy contents and there are repeated frames; instead of processing all of them and consuming time, it considers only frames show changes in the video; it is helpful and solves heavy processing problem [10] Model uses KATNA as keyframe extraction; it is open-source code written by python, and it does video extracting frames; it provides summary frames' of video content based on five elements are: LUV color space, degree of brightness, cluster of K-Means, Entropy or contrast filter and blur detection of extracted frames. It has been tested with different types of videos format [11]. As shown in Figure 3, it captures only 3 frames from a video, its duration is 6 seconds. The number of captured frames depends on the video duration and changes that appear within frames.

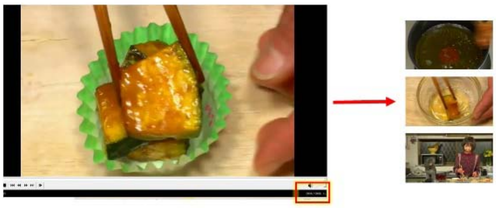
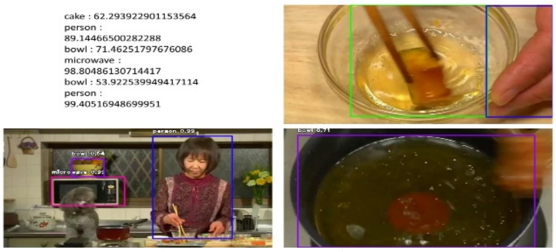


Figure 3: KATNA Result.

B. *Object Detection (using YOLO*

Detecting objects becomes an interesting subject in many areas; fast and accurate detecting is an important factor of any technique. The model needs it to construct sentences based on detected objects.

YOLO is one of the accurate and fast real-time detecting objects technique, it is based on predicting many objects, classifying the type of them and it shows accuracy percent. It does not slide the whole image and it is less error background detection than other deep learning techniques [12]. Figure 3 shows detected objects from YOLO in the model, it supports working with multiple images in one run and it takes 6-12 seconds to process an image based on CPU (it would run faster on GPU) [13].

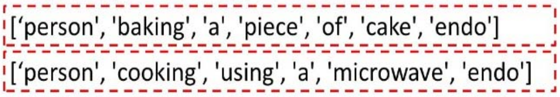


```
cake : 62.293922901153564
person :
89.14466500282288
bowl : 71.46251797676086
microwave :
98.80486130714417
bowl : 53.922539949417114
person :
99.40516948699951
```

Figure 4: YOLO Result.

C. *Sentence Generation based NLP (using LSTM):*

It is a process to construct sentences based on some words. LSTM is a type of RNN, it is back- forward model; which means its current result from the current layer depends on results from previous layers[14]. LSTM tries to relate between result words in a text file; it searches in trained sentences by scanning 3 words at a time in sentences, to find detected objects words; because LSTM is backward – forward technique, every result depends on the previous one. Then, it produces sentences showing the relationship between detected objects words to choose one of them as a video caption. As shown in Figure 5:



```
[‘person', 'baking', 'a', 'piece', 'of', 'cake', 'endo']
[‘person', 'cooking', 'using', 'a', 'microwave', 'endo']
```

Figure 5: LSTM Result.

We develop a model based on tensor flow and Keras, we use the Sequential model for NLP; which means, it is plain layers and takes\produces one input\output tensor [14]. It is an unsupervised model; it has been trained on created sentences by us. It contains – sentences, it has different types of objects. The sequential model has three types of layers are embedding, LSTM, and dense; we create six Sequential models contains the same types of layers. Each model search for a given word in created sentences. Embedding is the first layer, it converts integers to fixed-size vectors[15]. LSTM is an RNN type, it is a forward-backward technique. Dense connects LSTM layers, it implements activation function, and it is softmax activation[16]. Figure 6 shows the architecture of NLP:

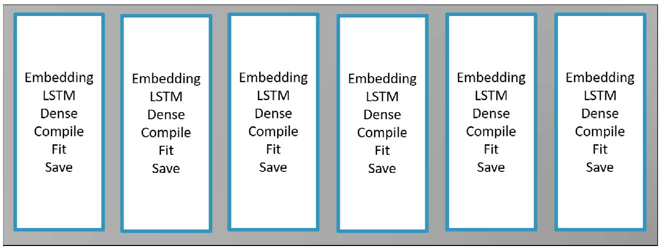


Figure 6: Architecture of LSTM.

IV. Discussion

Automatic video caption requires two important steps are: detecting objects and classifying their types (Encoder), plus step of generating sentence (Video Caption) based on detected objects as shown in Figure 1. To perform these steps, we propose model constructs of Key Frame Extraction (KATNA), object detection (YOLO), and generating sentences (LSTM).

At first, KATNA up to five frames based on video duration. It compares each consecutive two frames by KATNA elements as mentioned in the previous section, then it detects changes in frames within video and captures (frames (images)). After that, YOLO detects objects in captured frames by a surrounding box around objects with classification type plus percent accuracy; all this information is saved in a text file. Finally, LSTM reads detected objects words from the text file, then it generates an English sentence consists of subject and verb as a video caption.

V. Evaluation

Two evaluation methods implemented to measure the performance and accuracy of the model:

A. *the Metric for Evaluation of Translation with Explicit Ordering (METEOR):*

It has been used in many video captioning and description evaluation projects, especially, with short clips. It has 5 versions that calculate each hypothesis's alignment to its reference pair [17]. The proposed model has been evaluated by METEOR version 1.5, and the evaluation result of the proposed model is 0.35, it is a sufficient result. The following table shows a summary of some researchers who applied deep learning techniques to the MSVD dataset.

Table 1: Comparison with other Models.

Paper	**Percentage in METEOR on MSVD dataset**
Less Is More: Picking Informative Frames for Video Captioning [7].	0.33
M3: Multimodal Memory Modelling for Video Captioning [6].	0.2658
STAT: Spatial-Temporal Attention Mechanism for Video Captioning [2].	0.33
Temporal Deformable Convolutional Encoder-Decoder Networks for Video Captioning [3].	0.308
Proposed model	0.35

B. Human Evaluation

These criteria (Object Detection Quality, Sentence is Readable, Informativeness, Meaning preservation) have been formed in four questions and it has been sent to 20 persons for 3 different videos. Then, the average of the user's answers has been plotted.

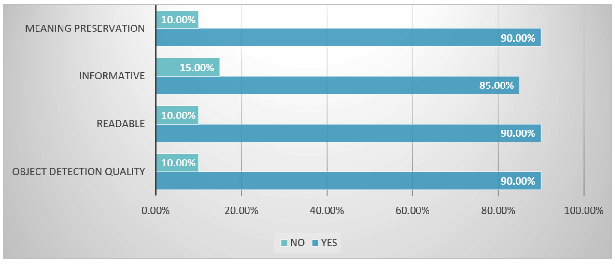


Figure 7: Human Evaluation Result.

VI. Conclusion and Future Work

In summary, video caption translates visual contents to text words sequentially (frame by frame and word by word), it is based on understanding video frames and transforms them to sentence (caption). Deep learning techniques have been greatly utilized in the field of video captioning research, which motivates researchers to develop a variety of video captioning framework which can automatically generate sentence (caption).

We develop a model using deep learning, which combines KATNA, YOLO, and LSTM. The model uses KATNA as a keyframe extraction, it chooses frames that show changes in video and remove redundant ones. The model applies YOLO on the MSVD dataset for object detection, and it generates English sentences using LSTM.

The model shows good accuracy and performance, it applies YOLO on the MSVD dataset, unlike the previous models that used other deep learning techniques. In the future, video captions will be measured by METEOR metric and human evaluation and compared with previous models using other deep learning techniques.

In the future, we think to train our model on more datasets; to be evaluated with different data. Also, we think to implement video caption-based Arabic language using (CAMeL Tools: An Open Source Python Toolkit for Arabic NLP). Finally, we think to evaluate the proposed model with different evaluation metrics such as BLEU, ROUGEL, and Diversity.

References

[1] . Li, B. Zhao, and X. Lu, ‘MAM-RNN: Multi-level Attention Model Based RNN for Video Captioning’, in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, Aug. 2017, pp. 2208–2214, doi: 10.24963/ijcai.2017/307.

[2] C. Yan *et al.*, ‘STAT: Spatial-Temporal Attention Mechanism for Video Captioning’, *IEEE Trans. Multimed.*, vol. 22, no. 1, pp. 229–241, Jan. 2020, doi: 10.1109/TMM.2019.2924576.

[3] J. Chen, Y. Pan, Y. Li, T. Yao, H. Chao, and T. Mei, ‘Temporal Deformable Convolutional Encoder-Decoder Networks for Video Captioning’, *ArXiv190501077 Cs*, May 2019, Accessed: Aug. 28, 2020. [Online]. Available: http://arxiv.org/abs/1905.01077.

[4] Chenyang Zhang and Yingli Tian, ‘Automatic video description generation via LSTM with joint two-stream encoding’, in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Dec. 2016, pp. 2924–2929, doi: 10.1109/ICPR.2016.7900081.

[5] R. A. Rivera-Soto and J. Ordonez, ‘Sequence to Sequence Models for Generating Video Captions’, p. 7.

[6] J. Wang, W. Wang, Y. Huang, L. Wang, and T. Tan, ‘M3: Multimodal Memory Modelling for Video Captioning’, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, Jun. 2018, pp. 7512–7520, doi: 10.1109/CVPR.2018.00784.

[7] Y. Chen, S. Wang, W. Zhang, and Q. Huang, ‘Less Is More: Picking Informative Frames for Video Captioning’, *ArXiv180301457 Cs*, Mar. 2018, Accessed: Aug. 28, 2020. [Online]. Available: http://arxiv.org/abs/1803.01457.

[8] ‘Rivera-Soto and Ordonez - Sequence to Sequence Models for Generating Video C.pdf’. Accessed: Feb. 01, 2021. [Online]. Available: http://cs231n.stanford.edu/reports/2017/pdfs/31.pdf.

[9] 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)*, Las Vegas, NV, USA, Jun. 2016, pp. 5288–5296, doi: 10.1109/CVPR.2016.571.

[10] M. K. Asha Paul, J. Kavitha, and P. A. Jansi Rani, ‘Key-Frame Extraction Techniques: A Review’, *Recent Pat. Comput. Sci.*, vol. 11, no. 1, pp. 3–16, Feb. 2018, doi: 10.2174/2213275911666180719111118.

[11] Alok, ‘Video Key Frame Extraction With katna’, *Medium*, Oct. 22, 2019. https://medium.com/@Aloksaan/video-key-frame-extraction-with-katna-11971ac45c76 (accessed Aug. 28, 2020).

[12] J. Redmon and A. Farhadi, ‘YOLO9000: Better, Faster, Stronger’, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[13] L. Zhao and S. Li, ‘Object Detection Algorithm Based on Improved YOLOv3’, *Electronics*, vol. 9, no. 3, Art. no. 3, Mar. 2020, doi: 10.3390/electronics9030537.

[14] K. Team, ‘Keras documentation: Embedding layer’. https://keras.io/api/layers/core_layers/embedding/ (accessed Mar. 09, 2021).

[15] ‘tf.keras.layers.Embedding | TensorFlow Core v2.4.1’, *TensorFlow*. https://www.tensorflow.org/api_docs/python/tf/keras/layers/Embedding (accessed Mar. 09, 2021).

[16] ‘tf.keras.layers.Dense | TensorFlow Core v2.4.1’, *TensorFlow*. https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense (accessed Mar. 09, 2021).

[17] “https://www.cs.cmu.edu/~alavie/METEOR/index.html#Publications.” .