Video Captioning using Keras

Amol pawar
Electronic and Telecommunication
Vishwakarma Institute of Technology, Pune
amol.pawar191@vit.edu

Sawan Damase
Electronic and Telecommunication
Vishwakarma Institute of Technology, Pune
sawan.damase19@vit.edu

Shubham Potphode
Electronics and Telecommunication
Vishwakarma Institute of Technology, Pune
Shubham.potphode19@vit.edu

Rushikesh Ugalmugale
Electronics and Telecommunication
Vishwakarma Institute of Technology, Pune
rushikesh.ugalmugale19@vit.edu

Prof. Suhas Bhise
Electronics and Telecommunication
Vishwakarma Institute of Technology, Pune
suhas.bhise@vit.edu

***Abstract*— Video captioning is the process of collecting information from a video and using those attributes to create video captions. The computer vision field faces a major difficulty in automatically generating natural language descriptions of videos. The use of 2-D and/or 3-D Convolutional Neural Networks (CNNs) to encode video material and Recurrent Neural Networks (RNNs) to decode a text has yielded the most recent development in this field. Long Short-Term Memory with Transferred Semantic Characteristics (LSTM) is a revolutionary deep architecture that integrates transferred semantic attributes learnt from images and videos into the CNN + RNN framework by training them end-to-end in this article.**

Keywords— Convolutional neural networks (CNN), Recurrent Neural Networks (RNNs), Video Captioning, Long Short-Term Memory(LSTM).

I. INTRODUCTION

Video captioning, also known as natural language video description, has been a significant problem to both the computer vision and language processing fields. This new idea has attracted a lot of attention from researchers. The advancement of artificial intelligence in domains like computer vision, natural language processing, and machine learning has sparked a surge in interest in complicated intelligence problems that require simultaneous processing of natural language and images. With the exponential growth in video data output, fueled by the popularity of online video sharing platforms like YouTube, Dailymotion, and Netflix, researchers' interest in autonomous video analysis has increased.

Video captioning and video question-answering are two common video-based complicated intelligence challenges. We hope to contribute to the existing research on video captioning algorithms with this work. The sequence learning method, on the other hand, uses sequence learning models to directly translate video input into a phrase, and is largely motivated by recent breakthroughs in machine translation utilising Recurrent Neural Networks (RNNs).

The concept is based on a translation encoder-decoder system.A 2-D/3-D Convolutional Neural Network (CNN) encoder scans a video and generates a vector of visual representations, which is then fed into a decoder RNN, which generates a natural phrase.While promising results have been published, sequence learning techniques based on CNNs and RNNs translate directly from video representations to language, leaving the high-level semantic clues in the video unexplored. Furthermore, high-level semantic information, i.e., semantic characteristics, has been proven to be useful in language task vision. Captions are especially useful for those who are watching movies in a language other than their own, children and adults learning to read, and people who are deaf or hard of hearing. When you make your content accessible, you make it possible for those who are deaf or hard of hearing to view your videos. Video cannot be crawled by search engines, but text can. If you want your videos to rank higher on Google, include captions and a transcript.

II. LITERATURE REVIEW

So as of now, there are a lot of methods proposed for Video captioning with Keras.So we will discuss existing systems in brief, first. There are many methods for Video captioning with Keras.Generally they are LSTM architecture by using deep learning based, Machine learning based.

The proposed video captioning model uses not only visual features, but also semantic features to effectively represent a video. Visual features are extracted using the existing ResNet (residual network) and C3D (3DConvNet) convolutional neural networks (CNN), while semantic features are extracted using novel semantic feature networks. Semantic features are divided into dynamic features.representing actions in the video and static semantic features representing objects, people, and background[3][1].

The encoder decoder framework for video captioning consists of two parts: one part to extract features representing a video using an encoder, and the other part to output word sequences of the caption using a decoder. In particular, CNN models such as pre-trained very deep convolutional networks (VGG), ResNet, and C3D have been used as encoders, while recurrent neural networks (RNN) have been mainly used as a decoder[1][4].

Author investigates how linguistic knowledge mined from large text corporations can aid the generation of natural language descriptions of videos. Specifically, They integrate both a neural language model and distributional semantics trained on large text corpora into a recent LSTM-based architecture for video description. We evaluate our approach on a collection of Youtube videos as well as two large movie description datasets showing significant improvements in grammaticality while modestly improving descriptive quality[2].

Long Short-Term Memory with Transferred Semantic Attributes (LSTM-TSA) a novel deep architecture that incorporates the transferred semantic attributes learned from images and videos into the CNN plus RNN framework, by training them in an end-to-end manner. The design of LSTM-TSA is highly inspired by the facts that Semantic attributes play a significant contribution to captioning, and Images and videos carry complementary semantics and, Thus can reinforce each other for captioning.[1][3] To boost video captioning, Author propose a novel transfer unit to model the mutually correlated attributes learn from images and videos.

LSTM-TSA achieves to-date the best published performance in sentence generation on MSVD: 52.8% and 74.0% in terms of BLEU@4 and CIDEr-D.[3] Superior results are also reported on M-VAD and MPII-MD when compared to state-of-the-art methods. propose a novel data-set which contains transcribed ADs, which are temporally aligned to full length HD movies.[4] In addition the author also collected the aligned movie scripts which have been used in prior work and compared the two different sources of descriptions. In total the MPII Movie Description data set (MPII-MD) contains a parallel corpus of over 68K sentences and video snippets from 94 HD movies. We characterize the data-set by benchmarking different approaches for generating video descriptions [5].We propose to use a neural-network-based framework to generate textual captions for the given input video. Our pipeline consists of three distinct stages[6]. Deep learning approaches, especially deep Con-evolution Neural Networks (ConvNets), have achieved overwhelming accuracy with fast processing speed for image classification. [8]. The state-of-the-art on video captioning benchmarks. Notably, even using a single network with only RGB stream as input, HRNE beats all the recent systems which combine multiple inputs, such as RGB ConvNet plus 3D ConvNet.

The first stage is feature extraction, wherein we extract both whole video and keyframe image based features from the input video. As the whole video based feature we use dense trajectories Keyframe image features are extracted by feeding these images through Convolutional Neural Net- works (CNN) trained on the ImageNet database[8].

III. PROPOSED METHOD

The flowchart for the proposed system is shown in Figure 1. Data collection is the first step, followed by feature extraction from video frames, cleaning, and preprocessing. We added the bos> and eos> tokens before and after each caption as the only text preparation we did. The model knows to start forecasting from here since bos> marks the beginning of the sentence, and eos> denotes the end of the statement, which is where the model knows to stop predicting. Most text generation problems are solved using an encoder-decoder design. What is a video, exactly? Isn't it possible to call it a sequence of images? For everything involving sequence, we always utilize RNNs or LSTMs. In our instance, an LSTM will be used.Loading data into the model now that we know the model is also a crucial element of the training. The model was trained for 150 epochs. One epoch's training takes roughly 40 seconds to complete. For Tesla T4 training, I utilised the free version of Colab.Then loading the dataset and at last we get captions.

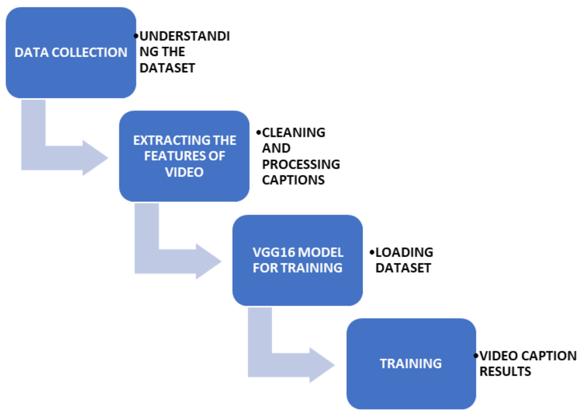

DATA COLLECTION
•UNDERSTANDING THE DATASET
EXTRACTING THE FEATURES OF VIDEO
•CLEANING AND PROCESSING CAPTIONS
VGG16 MODEL FOR TRAINING
•LOADING DATASET
TRAINING
•VIDEO CAPTION RESULTS

Fig.1 Flowchart of the system

IV. METHODOLOGY

A. Dataset:

We used Microsoft's MSVD data set for this investigation. The data set is available here. This data set includes 1450 short YouTube clips for training and 100 films for testing that have been hand labelled.
Each video has a unique ID, and each ID includes approximately 15–20 captions.

B. Understanding the Dataset:

The training data and testing data folders are located in the data set. Each folder has a video subfolder that contains the videos that will be used for both training and testing. There is also a feat subfolder in these folders, which stands for features. The video's features are contained in the feat folders. There are also json files for training label and testing label. The captions for each ID are contained in these json files. The following is how we can read the json files:

```
train_path='training_data'
TRAIN_LABEL_PATH=os.path.join(train_path, 'training_label.json')
# mentioning the train test split
train_split = 0.85
# loading the json file for training
with open(TRAIN_LABEL_PATH) as data_file:
   y_data = json.load(data_file)
```

C. Extracting Features of Video:

The number of frames taken will vary depending on the duration of the video. As a result, only 80 frames from each video are taken for the sake of simplicity. Each of the 80 frames is run through a VGG16 that has been pre-trained, and 4096 features are retrieved. The (80, 4096) shaped array is formed by stacking these characteristics. The number of frames is 80, and the number of features taken from each frame is 4096.

D. Captions Cleaning and Preprocessing:

Now we'll load all of the captions and pair them with their respective video IDs. Here's what I came up with. The train list contains two captions as well as the video ID. I added the <bos> and eos> tokens before and after each caption as the only text preprocessing I did.
The model knows to start forecasting from here since < bos> marks the beginning of the sentence, and <eos> denotes the end of the statement, which is where the model knows to stop predicting.

E. Model For Training:

An encoder-decoder architecture is the chosen model for most text generating challenges. We'll employ this sequence-to-sequence architecture in our problem statement since text must be generated. What exactly is a video? Isn't it possible to term it a picture sequence? We always use RNNs or LSTMs for anything that has to do with sequence. An LSTM will be used in our scenario. Let's look at the decoder now that we've decided to use LSTM for the encoder. Captions will be generated by the decoder. Because captions are just a series of words, we'll use LSTMs in the decoder as well.

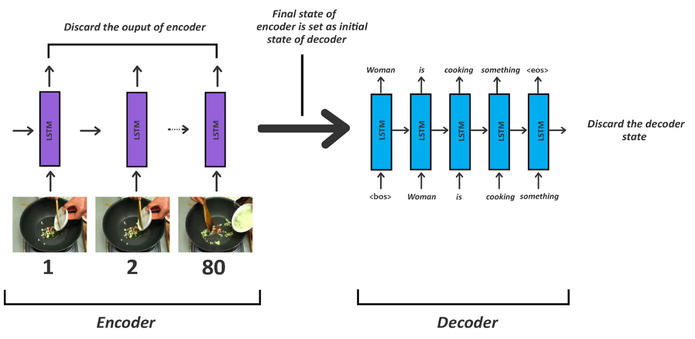
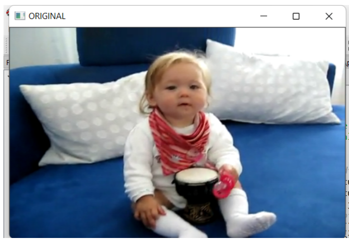
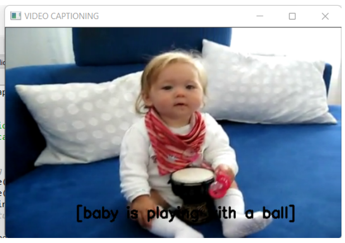

Discard the ouput of encoder
Final state of encoder is set as initial state of decoder
LSTM
Woman
is
cooking
something
<eos>
<bos>
Discard the decoder state
1
2
80
Encoder
Decoder

Fig.2 Training Model

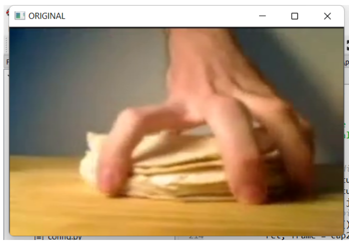
The first frame's features are supplied into the encoder's first LSTM cell in this image. This is followed by the second frame's features, and so on until the 80th frame. Because we're only interested in the encoder's end state for this challenge, all of the encoder's other outputs are ignored. The encoder LSTM's final state now serves as the decoder LSTM's initial state. The first decoder uses LSTM as an input to begin the sentence. Each word in the caption from the training data is supplied one by one until the total number of words in the caption is reached.
So, in the case of the example above, if the true caption is "lady is preparing something," the decoder begins with LSTM in the first decoder. The next word from the real caption appears in the next cell: woman is fed, followed by cooking something. This comes to a close with a token. The encoder's time steps are equal to the number of LSTM cells we'll use for the encoder, which is 80. The amount of features from video encoder tokens is 4096 in our situation. The decoder's time steps are equal to the number of LSTM cells in the decoder, which is ten, and the number of tokens is equal to the vocabulary length, which is 1500.

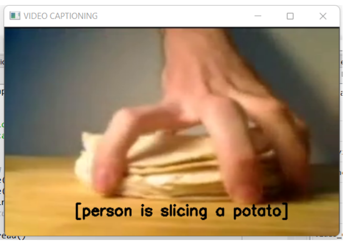
V. RESULTS



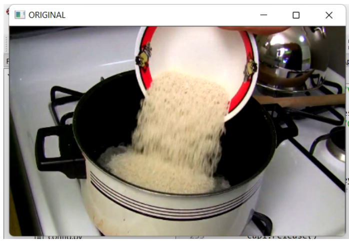


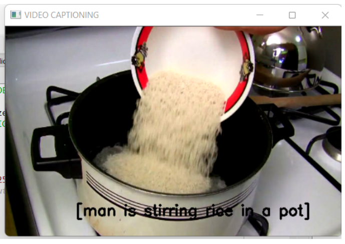
Original Video Frame video caption Frame





Original Video Frame video caption Frame





Original Video Frame video caption Frame

VI. CONCLUSION AND FUTURE SCOPE

In recent years, many models have been proposed and presented to generate captions for images and short videos. Although these models are helping to advance the technology, they suffer from inaccuracies due to fundamental constraints, resulting in limited use in practical situations.

This paper proposed a novel approach to video description. In contrast to related work, we construct descriptions using a VGG16 model. where frames are first read sequentially and then words are generated sequentially by using the LSTM network.This allows us to handle variable-length input and output while simultaneously modeling temporal structure. Our model achieves state-of-the-art performance on the MSVD dataset, and outperforms related work on two large and challenging movie-description datasets. Despite its conceptual simplicity, our model significantly benefits from additional data, suggesting that it has a high model capacity, and is able to learn complex temporal structure in the input and output sequences for challenging movie description datasets.

The system is evaluated with MSVD dataset and we get accuracy of 79.40 %.

Video captioning problem is not yet solved, as the best performance so far is still far from human-level captioning. Here, we identify several possible future directions, according to discussions in the literature and progress in related fields

VII. ACKNOWLEDGMENT

we would like to express our genuine and heartfelt gratitude to all those who have assisted us in our endeavour right from the start of this report without their active guidance help co-operation and encouragement we would not have made a headway in the project and report preparation we are grateful to prof.suhas bhise our project guide for his advice and support throughout the project from the literature review through the final implementation and interpretation of outcomes we are immensely obliged to him for his elevating inspiration encouragement guidance and kind supervision in the completion of our project.

VIII. REFERENCES

1. Y. Yu, H. Ko, J. Choi, and G. Kim, "End-to-end concept word detection for video captioning, retrieval, and question answering," in Proceedings of IEEE Conference on Computer Vision.
2. Subhashini Venugopalan UT Austin vsub@cs.utexas.edu; Lisa Anne Hendricks UC Berkeley lisaanne@berkeley.edu; Raymond Mooney UT Austin mooney@cs.utexas.edu; Kate Saenko Boston University saenko@bu.edu.
3. Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei University of Science and Technology of China, Hefei, China Microsoft Research, Beijing, China panyw.ustc@gmail.com, {tiyao, tmei}@microsoft.com, lihq@ustc.edu.cn
4. Anna Rohrbach; Marcus Rohrbach; Niket Tandon; Bernt Schiele;
5. Max Planck Institute for Informatics, Saarbr. ucken, Germany UC Berkeley EECS and ICSI, Berkeley, CA, United States
6. Rakshith Shetty and Jorma Laaksonen Department of Computer Science Aalto University School of Science P.O.BOX 15400, FI-00076 AALTO Espoo, Finland.
7. Kyunghyun Cho Bart van Merrienboer Caglar Gulcehre; Dzmitry Bahdanau Jacobs University, Germany;Fethi Bougares Holger Schwenk ;Yoshua Bengio Universite du M
8. Pingbo Pan; Zhongwen Xu; Yi Yang Fei Wu; Yueting Zhuang; Zhejiang University of Technology Sydney. {lighnt001,zhongwen.s.xu}@gmail.com yi.yang@uts.edu.au {wufei,yzhuang}@cs.zju.edu.cn
9. Bairui Wang;Lin Ma; Wei Zhang; Wei Liu; Tencent AI Lab School of Control Science and Engineering, ShandongUniversity
10. Word Beam Search: A CTC DecodingAlgorithm "https://towardsdatascience.com/word-beam-search-a-ctc-decoding-alg orithm-b051d28f3d2e"
11. Islam, Saiful & Dash, Aurpan & Seum, Ashek & Raj, Amir & Hossain, Tonmoy & Shah, Faisal. (2021). Exploring Video Captioning Techniques: A Comprehensive Survey on Deep Learning Methods. SN Computer Science. 2. 10.1007/s42979-021-00487-x.