International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 12 | Dec 2020 www.irjet.net p-ISSN: 2395-0072

Real Time Video to Text Summarization using Neural Network

Abhishek Yadav[1], Anjali Vishwakarma[2], Shyama Panickar[3], Prof. Satish Kuchiwale[4]

[1-3]Students, Dept. of Computer Science, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India

[4]Teacher, Dept. of Computer Science, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India

---

Abstract - *This paper represents a model for automatically identifying the important parts of a real time video and annotating the video with captions to enable a rich and more concise condensation of the video. The main idea is to select the key frames from the live video feed, and caption them for text summarization. On the other hand, captioned image can help a video captioning model to learn better semantic representations. This model proposes a general neural network configuration that jointly considers two supervisory signals in the training phase and generates both video summary and corresponding captions for a given images in the test phase.*

Key Words: Caption signals, video captioning model, Semantic representations.

1. INTRODUCTION

The prevalence of recording devices encourages more people to capture their daily life with video data content. But, the large amount of video data makes it more difficult to navigate, particularly long videos such as surveillance videos or CCTV footages. For larger videos, automatically identifying the important parts/frames of the video content and enabling them with captions will give a richer and more concise condensation of the video. It is still time-consuming for users to navigate or to search through a summarized video. So, the automatic video summarization has been proposed to extract a compact representation of the video data into textual form. The proposed system offers a brief semantic understanding of a long video just through a text summary.

2. RELATED WORK

2.1 Video Description

Traditionally, researchers use unsupervised methods for automatic real-time video summarization. This is done either by taking a holistic view of the entire video or by identifying the local differentiation among the adjacent frames. In these methods, researchers design criteria such as relevance, diversity, and representativeness to select important frames or shots from the video. Some researchers utilize web media and metadata as prior knowledge to generate better summarization results. Visual attention is also used to select important frames. However, video summarization requires a semantical understanding of the video content and is hard to model with a heuristic design. Our system also uses a supervised method with RNN for video summarization I.e. LSTM. Different from previous work, we incorporate the context of a video captioning model to learn a more semantic- driven video representation for the summarization task.

2.2 Image Captioning

Video understanding is an important subject in the field of computer vision. Researchers have worked on many different topics of video content analysis such as video classification, action recognition, and video captioning. Recently, with the success of image captioning using CNN and RNN models, many researchers have adopted similar approaches to describe video content with natural language. There are methods that emphasize caption generation including:

Semantic supervision which designs auxiliary objectives to exploit visual semantic concepts to improve captioning quality.

(2) Approaches to mitigate the objective mismatch problem.

3. METHODOLOGY

In this section, we have discussed the methodology for real-time video to text summarization using deep neural network-based models. We have introduced a model for feature extraction and image captioning. The initial phase is the selection of the keyframes from the real-time video feed at every interval. For the image feature extraction, we have used the Convolutional Neural Network (CNN). To train the model on the Flickr 8k dataset first we have to extract the features from the images by using the inceptionxv3 model and then we have to extract the captions for the images also. The captions will need to be converted to numbers before presenting to the model. The first step in encoding the captions is to create a consistent mapping

from words to unique integer values. The model will be provided with one word & the image and it generates the next word. Then the first two words of the caption will be provided to the model as input with the image to generate the next word and so on. This is how the model will be trained. Now the selected keyframes from the real-time video feed are stored in one dictionary in jpeg or png form. Extract the features of all the images present in the dictionary and load the tokenizer folder and the trained model. After loading the model pass all the image features to it for the image captioning and store the caption and image as a result.

3.1 Algorithm

CNN Stands for Convolutional Neural Networks. CNNs are designed to map image data to an output variable. They have proven so effective that they are the go-to method for any sort of prediction problem involving image data as an input. The advantage of using CNNs is their ability to develop an inside representation of a two- dimensional image. It allows the model to find out the position and scale-invariant structures within the data, which is vital when working with images.

Uses of CNN in image captioning :

1. Image data
2. Classification prediction problems
3. Regression prediction problems More generally, CNNs work well with data that features a spatial relationship. Although not specifically developed for non-image data, CNNs achieve state-of-the-art results on problems like document classification (For example, there's an order relationship between words during a document of text) utilized in sentiment analysis and related problems.

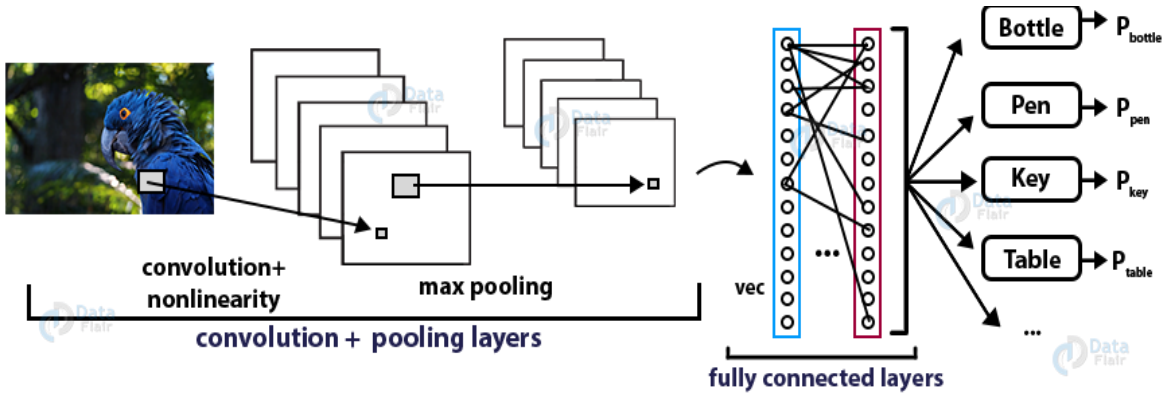

Fig-1: Convolutional Neural Networks (CNN)

RNN stands for Recurrent Neural Networks. RNNs generally and LSTMs especially have received the foremost success when working with sequences of words and paragraphs, generally called tongue processing. This includes both sequences of text and sequences of speech represented as a statistic. They are also used as generative models that need a sequence output, not only with text, but on applications like generating handwriting.

Uses of RNN in image captioning:

1. Text data
2. Speech data
3. Classification prediction problems
4. Regression prediction problems
5. Generative models

So now CNN acts as a feature extractor that compresses the knowledge within the original image into a smaller representation. Since it encodes the content of the image into a smaller feature vector hence, this CNN is usually called the encoder. When we process this feature vector and use it as an initial input to the subsequent RNN, then it might be called decoder because RNN

would decode the method feature vector and turn it into natural language. Traditionally, an easy strategy for modeling sequence is to map the input sequence to a fixed-sized vector using one RNN, then to feed the vector to a SoftMax layer for classification or other tasks.

LSTM stands for Long short-term memory network. LSTM is a kind of recurrent neural network. LSTM was proposed by [Hochreiter and Schmidhuber, 1997] to specifically address this issue of learning long-term dependencies. The LSTM maintains a separate memory cell inside it that updates and exposes its content only deemed necessary. A number of minor modifications to the standard LSTM unit have been made. While there are numerous LSTM variants, here we describe the implementation employed by Graves. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long term memory but can give more accurate predictions from the recent information. LSTM can by default retain the information for long period of time. It is used for processing, predicting and classifying on the basis of time series data.

Applications of LSTM :

1. Language Modelling
2. Machine Translation
3. Image Captioning
4. Handwriting generation
5. Question Answering Chatbots

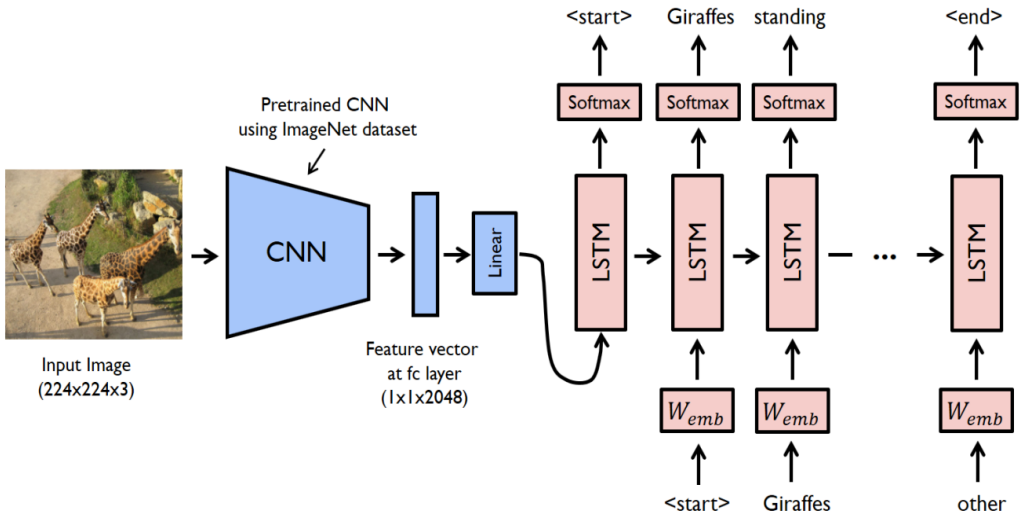

Fig-2 : CNN + LSTM for Image Captioning

3.2 Approaches of Model

On the ImageNet dataset Inception v3 is a widely-used image recognition model. Inceptionv3 is a convolutional neural network for assisting in image captioning and object recognition. The inceptionv3 model is the collection of many ideas developed by multiple researchers over the years. The model itself is made up of symmetric and asymmetric building blocks, including average pooling, convolutions, concatenation, dropouts, max pooling, and fully connected layers. The current application of Inception v3 is right at the edge of being input-bound. Images have to be redeemed from the file system, decoded, and then preprocessed. Different types of preprocessing stages are available, ranging from moderate to complex. If we use the most complex of preprocessing stages, the large number of expensive operations executed by the preprocessing stage will push the system over the edge and the training pipeline will be preprocessing bound. However, it is not at all necessary to resort to that level of complexity to attain greater than 78.1% accuracy, and we instead use a moderately complex preprocessing stage that tilts the scale in the other direction.

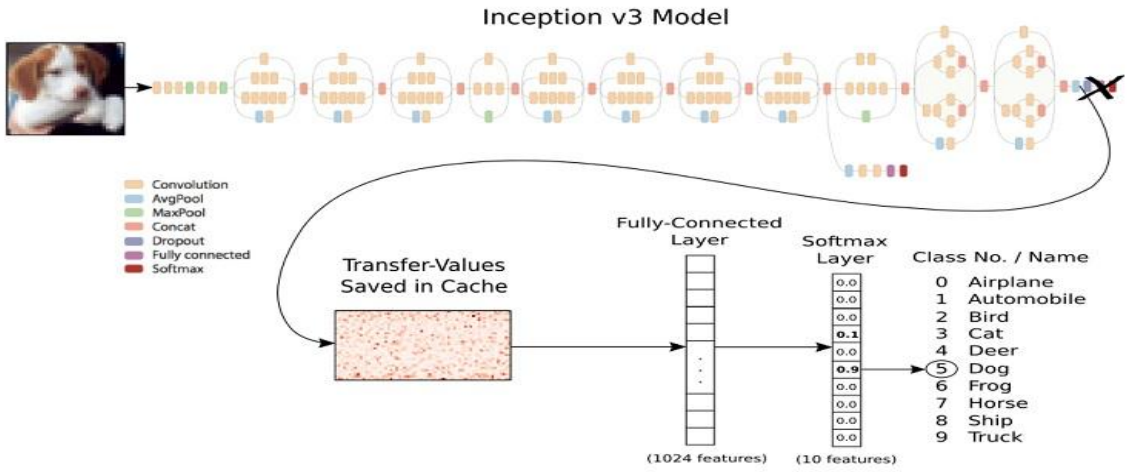

Fig-3 : Inception V3 Model

We have used 2 methods for predicting the captions. Argmax Search and Beam Search. Argmax Search is where the maximum value index in the 8256 long predicted vector is extracted and appended to the result. This is done until we hit <end> or the maximum length of the caption. Beam Search is where we take top k predictions, feed them again in the model and then sort them using the probabilities returned by the model.

VGG-16 model is mainly used for the image classification and image detection. This pretrained model using convolutional neural network(CNN) for large scale image recogniton. We have used Flicker 8K dataset to train the model for generation of caption. All the images in the dataset are resize, resample and croped to 256 x 256 resolution for the resulting image. As shown in the figure 4 there are different layers in the model for detecting the image and generate the caption. The image is passed through this layers the first layer is convolution and ReLU which is used for the filtering the image, the second layers is max pooling it performs a 2 x 2 pixel window with stride 2. There is total 16 hidden layers in this model. Long-short term networks (LSTM) are distinct kinds of RNNs which are opted when a large sequence of information needs to be sustained. For this reason, we have used the LSTM with a soft attention model for our language model design. The LSTM structure has an inside memory unit called a cell (ct), several gates associated with units, and a hidden layer output (ht).

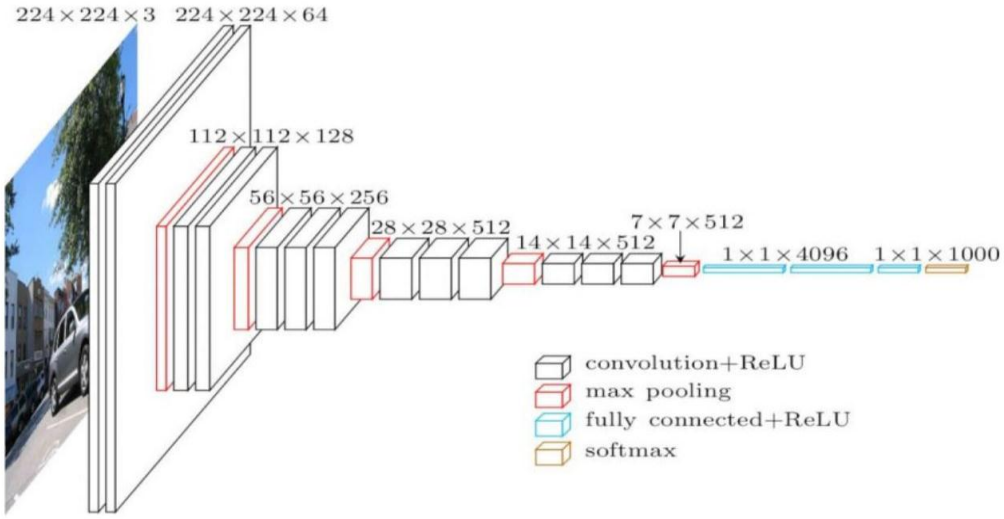

Fig-4 : VGG-16 Model

3.3 System Design

1) Flowchart of the Proposed System

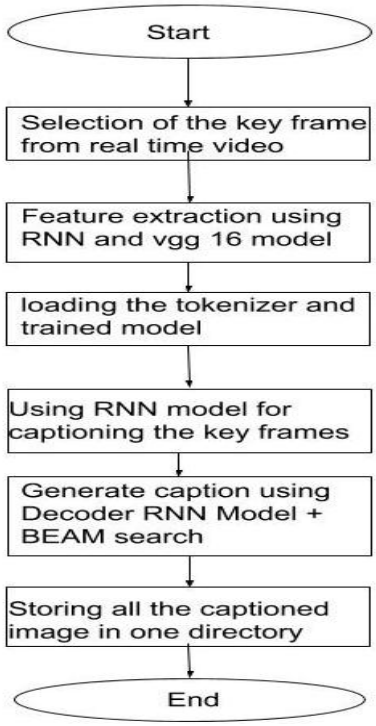

Chart-1 : System Flowchart

2) Structure of the Model

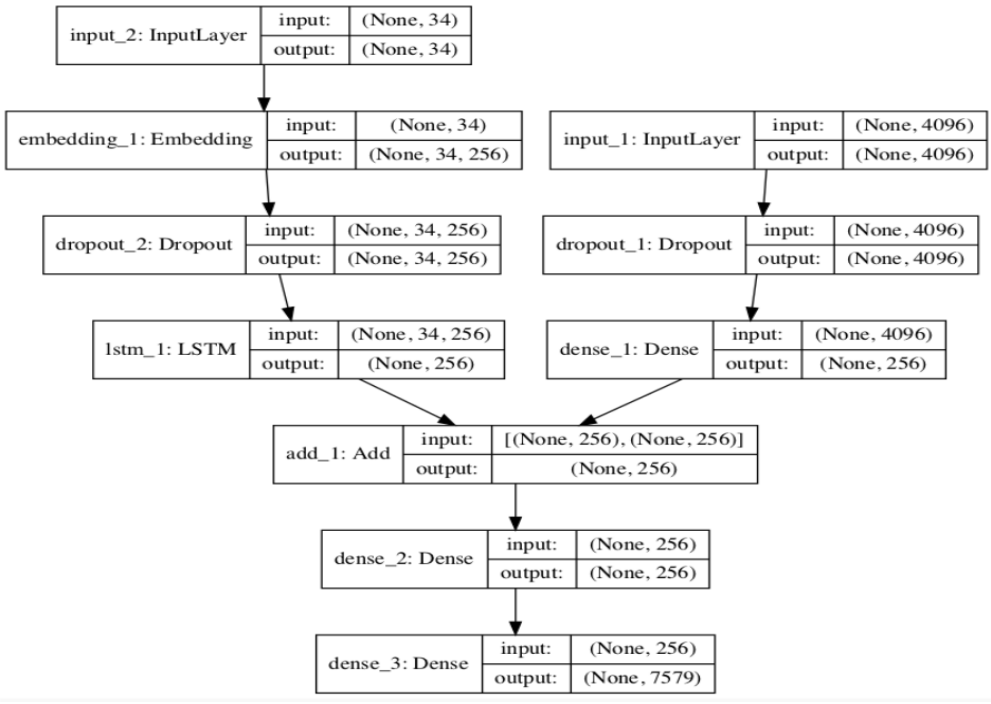

Fig-5 : Structure of the model

3) Use Case Diagram

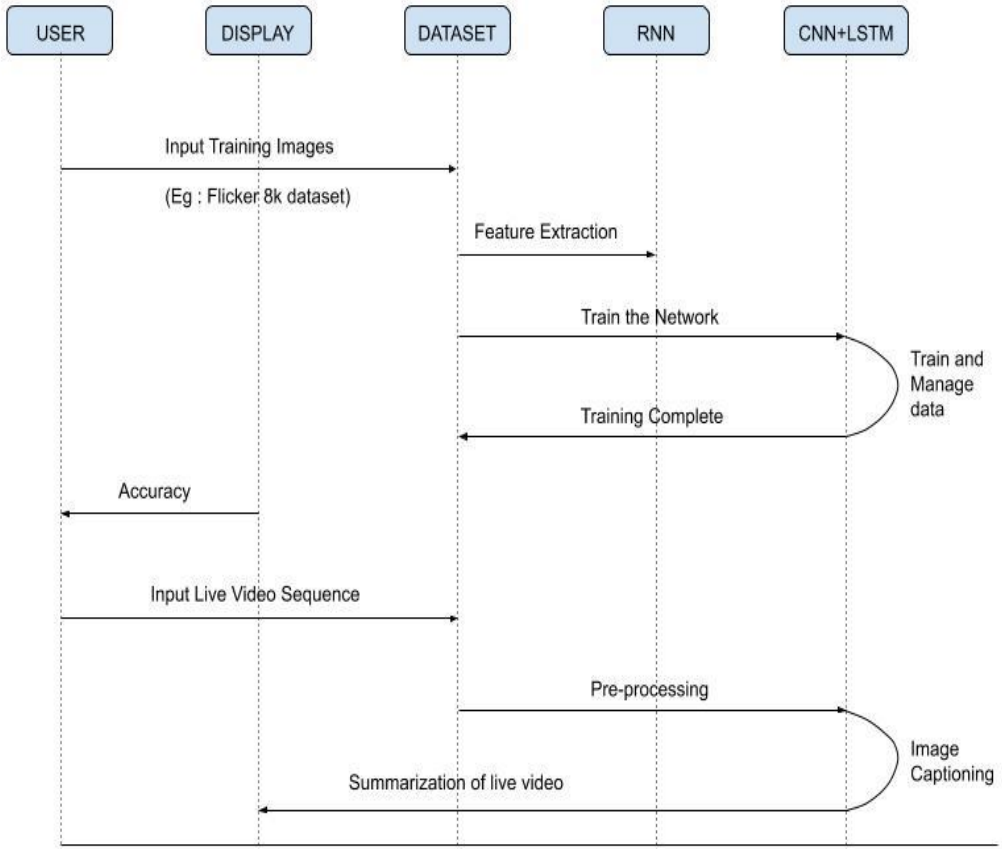

Fig-6 : Use Case Diagram

4. RESULTS

We have use Flicker 8K dataset for training the model. It contains two folder Flicker8K_Data and Flicker8K_text. The data folder contains 8092 images in JPEG format with different shape and size. For training Purpose we have used 6000 images and 1000 for testing the accuracy of the model and remaining for the validating the model. The test folder contains all the captions for the images with their image name and the caption folder is already split into different file like test_captions, train_captions, etc. The model has been train on the GPU using Tensorflow and Keras. The figure 7, 8, and 9 shows the text summarization of the images which are selected as the key frame from the live video feeds for the image captioning.

Fig-7 : Result 1 **Fig-8 :** Result 2

Fig-9 : Result 3

To train the model on the flickr 8k dataset first we have to extract the features from the images by using the inception v3 model and then we have to extract the captions for the images also. Every image has at least 5 captions and those captions are stored in the dictionary form. All the captions are saved into caption.txt in the form of 'id' 'caption' one per line. Example : 225285_487f2 stadium full of people watch the game. We have encoded the caption and trained the encoded caption for generation of text. Tokenizer class which is provided by keras is used for the mapping of the encoded caption.

X1	X2(text sequence)	y(word)
image	startseq,	little
image	startseq, little,	girl
image	startseq, little, girl,	running
image	startseq, little, girl, running,	in
image	startseq, little, girl, running, in,	field
image	startseq, little, girl, running, in, field,	endseq

Fig-10 : Caption Sequence

As shown in fig 10 each and every captions gets split into single word and the image which is provided to the model for training. This is how model will be trained. For example, the input sequence "little girl running in field" would be split into 6 input-output pairs to train the model. The inception v3 model is trained on the flickr 8k dataset for 20 epochs. The best result is train_loss = 2.4566 and the val_loss= 3.0870. The inception v3 model has been improved 9.8526 to 2.4566 for train_loss and 10.6548 to 3.0870 for val_loss, after 9th epochs the inception v3 model didn't improve the train_loss and val_loss.

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 34)	0	
input_1 (InputLayer)	(None, 4096)	0	
embedding_1 (Embedding)	(None, 34, 256)	1940224	input_2[0][0]
dropout_1 (Dropout)	(None, 4096)	0	input_1[0][0]
dropout_2 (Dropout)	(None, 34, 256)	0	embedding_1[0][0]
dense_1 (Dense)	(None, 256)	1048832	dropout_1[0][0]
lstm_1 (LSTM)	(None, 256)	525312	dropout_2[0][0]
add_1 (Add)	(None, 256)	0	dense_1[0][0] lstm_1[0][0]
dense_2 (Dense)	(None, 256)	65792	add_1[0][0]
dense_3 (Dense)	(None, 7579)	1947803	dense_2[0][0]

Total params: 5,527,963
Trainable params: 5,527,963
Non-trainable params: 0

Fig-11 : Structure of Model and Shape of the Layers

The structure of the model is shown in fig 11 it also shows the shapes of the hidden layer in the model.

For example its shows the total trainable parameters are 55,27,96.

The VGG-16 model is trained on the flickr 8k dataset for 15 epochs. The best result is train_loss = 2.5575 and the val_loss = 3.1779. The inception v3 model has been improved 9.8526 to 2.4566 for train_loss and 10.6548 to 3.0870 for val_loss, after 8th epochs the vgg-16 model didn't improve the train_loss and val_loss. The trained model validates on the 1000 images to check the accuracy. Now the selected keyframes from the real time video feeds are stored in one dictionary in jpeg or png form. Extract the features of all the images present in the dictionary and load the tokenizer folder and the trained model. After loading the model pass all the image features to it for the image captioning and store the caption and image as a result.

The BLEU score is used for the measure that evaluate skill of the model. Below table shows the BLEU score of the model on the test dataset.

Table-1 : BLEU Score on Test Images

BLEU - 1	0.401 to 0.578
BLEU - 2	0.176 to 0.390
BLEU - 3	0.099 to 0.260
BLEU - 4	0.059 to 0.170

5. CONCLUSION

Hence , we conclude with a joint end-to-end model, which uses deep neural network to generate natural language descriptions and abstractive text summary of a live input video sequence. We have used the dataset containing images and captions for training the model. This model can be implemented on the live video feeds like cctv footages or the offline videos. We are selecting the key frames from videos and those key frames are used for image captioning to generate the text summary. This textual conversion not only reduces the size of video data but also enables users to index and navigate information through it. It shows that both our models execute well on standard datasets. This representation will not only save processing time but will also save storage space.

6. REFERENCES

[1] Bor-Chun Chen, Yan-Ying Chen, Francine Chen: "JOINT VIDEO SUMMARIZATION AND CAPTIONING" in University of Maryland College Park, Maryland, USA, FX Palo Alto Laboratory, Inc. Palo Alto, California, USA.

[2] Pengfei Liu, Xipeng Qiu, Xuanjing Huang "Recurrent Neural Network for Text Classification with Multi-Task Learning" in Shanghai Key Laboratory of Intelligent Information Processing, Fudan University School of Computer Science, Fudan University 825 Zhangheng Road, Shanghai, China

[3] Christian Szegedy, Vincent Vanhouck, Sergey Ioffe, Jonathon Shlens "Rethinking the Inception Architecture for Computer Vision" in Zbigniew Wojna University College London.

[4] Huda A. Al-muzaini, Tasniem N. and Hafida B. (2018) "Automatic Arabic Image Captioning using RNNLSTM-Based Language Model and CNN", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 9, No.6.

[5] San Pa Pa Aung, Win Pa Pa, Tin Lay Nwe "Automatic Myanmar Image Captioning using CNN and LSTM-Based Language Model" in Visual Intelligence Department, Institute for Infocomm Research, Singapore.

[6] Rahul, S. and Aayush, S. (2018). "Image Captioning using Deep Neural networks".

[7] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. (2017) Boosting image captioning with attributes. [Online]. Available: https://openreview.net/pdf?id=BkdpaH9ll

[8] https://www.researchgate.net/publication/339494679_Generation_of_Image_Captions_Using_VGG_and_ResNet_CNN_Models_Cascaded_with_RNN_Approach.

[9] B. Chen, Y.-Y. Chen, and F. Chen, "Video to text summary" in Proc. Brit. Mach. Vis. Conf. (BMVC), Sep. 2017, pp. 1–14.

[10] S. Chopra, and A. M. Rush, ‘‘Abstractive sentence summarization’’ in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol., Jun. 2016, pp. 93–98.

BIOGRAPHIES

“Abhishek Yadav Student Of Smt. Indira Gandhi College Of Engineering, in Computer Science branch, 2018-2021.”

“Anjali Vishwakarma Student Of Smt. Indira Gandhi College Of Engineering, in Computer Science branch, 2017-2021.”

“Shyama Panickar Student Of Smt. Indira Gandhi College Of Engineering, in Computer Science branch, 2017-2021.”

Prof. Satish Lalasaheb Kuchiwale is working as an Assistant Professor in Computer Engineering department in Smt. Indira Gandhi College of Engineering, Ghansoli, Navi Mumbai, affiliated to Mumbai University and having about 13 yrs. of experience. He has completed his M.E. in Computer Engineering.