International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 8 Aug 2021, pp: 1447-1457 www.ijaem.net ISSN: 2395-5252

AI Based Video Summary and Caption Generator

Madhamsetty Charitha[1], Pragati Kumari[2]
[1.] *R.V. College of Engineering Bengaluru ,India*
[2.] *R.V. College of Engineering Bengaluru ,India*

Submitted: 10-08-2021 Revised: 22-08-2021 Accepted: 25-08-2021

ABSTRACT— In recent times, summarizinga video and generating a caption for the video are examined as two individual tasks. Automatically generating short summarized video for a given video and giving a caption to it would reduce the storage requirements and provide easy understanding of the video. In this paper, we provide a solution for automatic video summarization and video captioning. Given a video, the goal is to generate a short video by selecting the key frames and to generate an appropriate caption for the video. Most of the approaches for video summarization used recurrent networks, but we use sequential fully convolutional networks. For video captioning, we use LSTMs since they have exhibited state-of-the-art performance in generating captions for an image.We train the video summarization model on a benchmark dataset TVSum and validate on another benchmark dataset SumMe. We train the video captioning model on sequence of frames to caption pairs so that it learns to relate a sequence of words to a sequence of frames on the MSVD dataset. We evaluate video captioning models taking videos which are not used for training in the video. Finally, we integrate both the models to generate both summarized video and caption for the input video.

Keywords— Fully convolutional Neural Networks, Long Short Term Memory Networks, Video Summarization, Feature extraction, Semantic segmentation, Video Captioning, sequence to sequence modelling, Encoder-Decoder network.

I.INTRODUCTION

Nowadays, videos are the easy and quick way of spreading the content. Since, videos are the best way to capture interest of people, they are widely used in social media. Also, online lectures have become popular these days. For every minute, almost 300 hours of videos are uploaded to Youtube[16]. Amazingly, 400 crores of videos are watched on youtube every day[16]. This shows that the people are using videos widely for different purposes. Generally, videos need more space to store on any device. So video storage is a challenge. The large quantity of data makes it difficult to examine and navigate, especially long videos like surveillance videos. The major challenges are based on the lack of density and grammar in the content of the video. As a result, summarizing a video by reducing the duration of the video as well as conveying the entire subject in the video has become one of the trending research topics. A perfect video summarization is that which gives users almost all details of an original video in a minimum amount of time.

The deep learning approaches to automatic video summarization are able to achieve state-of-the-art performance. But the major challenge is that processing a video and generating a summarized video involves complex computational tasks and it takes a long time to do so. Both supervised and unsupervised learning methods are proposed for video summary generation. In supervised methods, the model learns from the original videos and the ground truth summary videos. But in this approach, collecting data is difficult and processing takes more time. Unlike supervised learning, unsupervised learning does not use ground truth annotations. Yet, the reduction of frames takes place through clustering methods. But the accuracy achieved is less through unsupervised learning compared to supervised learning.

The summary generated can be of two types. 1. Keyframes and 2. Dynamic video summaries which comprises shots selected based on similarity among all shots. Here we use supervised learning to train a fully convolutional sequence network to generate the summary as a set of key frames as shown in figure 1.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 8 Aug 2021, pp: 1447-1457 www.ijaem.net ISSN: 2395-5252

The video captioning is also one of the trending research topics nowadays. Generating a description or caption for a video would give an idea of what the video is about which helps the user to decide upon whether to watch the video or not. Also it helps in specifying the genre of the video content which makes browsing videos based on their genre easier. Sequence to sequence modelling is one of the proposed methods for video captioning where sequence of frames are mapped to sequence of words thus forming sentences. Here, we use encoder-decoder LSTM for modelling the task of generating a caption for the video. The dataset we use for captioning contains short video clips which makes training and evaluation faster as processing of videos does not take longer time. Our motivation behind this work is that video summarization and captioning jointly are not proposed in the studies so far. This would help in performing two major tasks related to videos using the same system. Also, videos need not be processed twice which saves time.

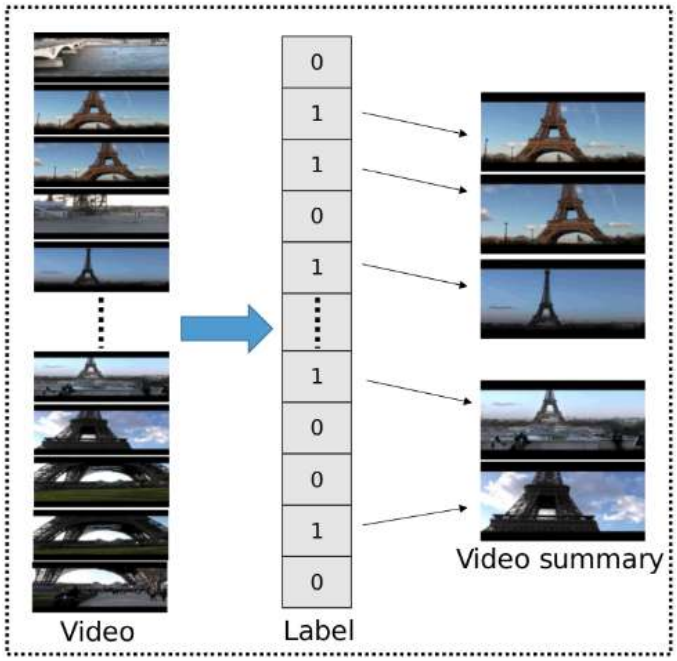

Figure.1. Selection of keyframes from the video. The labels are given to each frame by the model upon which the frame to be selected as keyframe or not depends[13].

II. RELATED WORK

Here, a few of the works related to video summarization and captioning are analyzed to understand what the existing systems offer and how they work.

Authors in [3] proposed a design of framework for a multi-faceted video summarization that extracts keyframes and entity summaries. Here, A video summarizer is developed using Caffe and DarkNet. As a result it is found that, when a video is given as an input, a video summary is obtained by extracting key frames at an entity level. The preprocessing of a video takes around 30 mins for a 2 hour video. Authors in [5] proposed an approach based on machine learning for video summarization. Here, it is found that semi-supervised learning is used to acquire the high level semantics but the system is trained only for summarizing home party and soccer videos. Authors in [9] introduced a design for summarization of a video using frame extraction and taking out the redundant frames by implementing feature extraction and classification techniques. It is found that video summary is initiated using k-means clustering and bayesian model and the results are compared but here the facial recognition and behaviour detection are not consolidated in the video summary .

Authors in [2]proposed a query focused video summarization that examines text queries.As a result, a new approach is found that takes user queries in account and initiates the personalized summary but the model is unable to extract key features at entity level such as things , objects etc.Authors in [7] introduced a design for query conditioned three player generative adversarial network for a query conditioned summarization of a video. It builds a query conditioned video

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 8 Aug 2021, pp: 1447-1457 www.ijaem.net ISSN: 2395-5252

summarization model using three player loss but here the trivial and short sequences are not generated for all the videos. Authors in [1] introduces GAN based video summarization and studies the benefits and constraints of using supervised and unsupervised reinforcement learning. It is found that GAN based training framework is a neural network which contains two adversarial networks and that combines the benefits of unsupervised and supervised learning but here the exact accuracy of using each model is not presented.

Authors in [4] proposed a design of end-to-end reinforcement learning based framework and to formulate summarization of a video as a sequential decision making process and to implement a deep summarization network to summarize videos.Here, a model is developed based on a label-free reinforcement learning algorithm to tackle unsupervised video summarization.However, the model could not enhance their performance i.e it produced the same accuracy as that of other unsupervised alternatives.Authors in [6] proposed a deep summarizer network to reduce distance between training videos and a distribution of their summarizations.As a result, The deployment of an efficient unsupervised video summarization with LSTM networks but there are some failed cases occurred in videosthat consists of frames with the slow motions but there is no scene change.

Authors in [8] introduces a design of hierarchical reinforcement learning framework for captioning of a video. As a result, they built a video captioning model which automatically provides a textual illustration of the actions in a video but the preprocessing of video takes a long time. Authors in [10] introduces a deep learning framework for the video captioning.It summarizes the evaluation results of captioning method and it is found that the principal component analysis is the best performer but here it could not provide the detailed analyses of each method of video captioning.

III. METHODOLOGY

We formulated our objectives as two separate tasks i.e video summarization and video captioning. Figure 2 shows the high level working of our system.

A.Video Summarization

Given an input video, the aim is to automatically generate a summarized video which conveys the entire story in the video and the duration of the summarized should not exceed 15% of the duration of the original video.

The dataset used for training is the standard TVSum dataset[17] which contains 50 videos of 2 to 6 minute duration along with the 20 ground truth annotations for each video in the form of shot-level importance scores i.e how likely each shot (sequence of frames) to be in the summarized video. The dataset used for validation is SumMe dataset[18] which is also a standard dataset for video summarization. It contains 25 videos each of which is annotated with 15 human summaries.

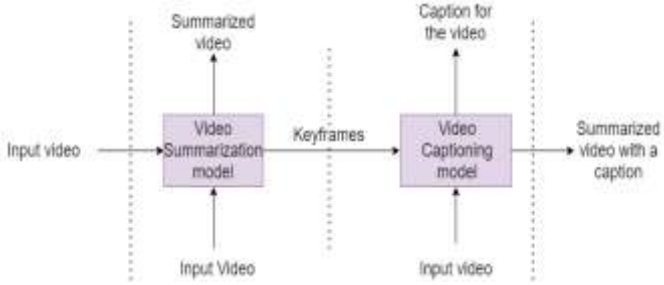

Figure.2. High level architecture of our system. It has two separate systems one for video summarization and another for captioning. For a given input video, our system can generate a summarized video or caption for the video or caption along with summarized video depending upon the user's choice.

The task of video summarization is organized as 1. Video processing into frames 2. Feature extraction of each frame 3. Semantic segmentation o4. Generation of summarized video. Figure 3 shows the flow chart of our methodology.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 8 Aug 2021, pp: 1447-1457 www.ijaem.net ISSN: 2395-5252

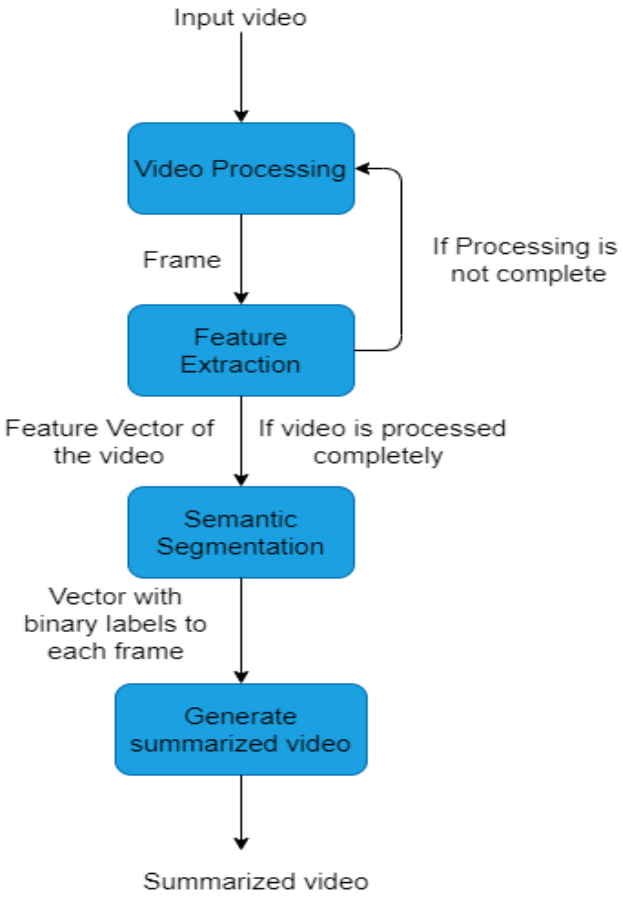

Figure.3. Flowchart showing the methodology in our video summarization model.

For video summarization, we use pytorch for building, training and validating the model.Feature extraction is done using the pre-trained version of GoogleNet model trained on ImageNet dataset[11]. The main reason to use GoogleNet for feature extraction is that it is 22 layers deep and has achieved the lowest error rate as compared to other networks for image classification. All the children but the last two are in the feature extraction part of the model. Features extracted from these layers are stored in an array.

We preprocess the dataset i.e we create a preprocessed file for easy processing of the videos and their ground truth summaries. The created file contains number of frames, feature vector of shape (320,1024), scene change points which is of shape (number of segments, 2 i.e start and end of each segment), number of frames in each segment, user summary which is of shape (20, number of frames in the video) i.e it contains 20 binary vectors and each vector contains 0s and 1s- 0 represent that the frame is not in the summarized video and 1 represent that the frame is in the summarized video. Scene change points and the binary vectors which are created from shot-level importance scores of the dataset are taken from [4].

The preprocessed data also contains labels to each frame. The label to each frame is nothing but 0 or 1 representing whether the frame is keyframe or not. Since there are 20 user summaries, the label to each frame is calculated by the following method. Initially, all the 20 user summaries to each frame are summed up and the frame with higher sum is given higher priority. Then we take each frame according to the sorted array, then we add all the 20 user summaries of that particular frame and the sum is stored in an array named sum_arr. Also, the summary (0/1) to all the frames is summed up for each user and such 20 sums are stored in true_sum_array. Then we calculate 'frscore' to each frame considering their priority using the following calculations. Initially, the best frscore would be zero and frame_count is 1. For each entry i in true_sum_array and j in sum_arr,

precision = j/(frame_count+1e-8)

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 8 Aug 2021, pp: 1447-1457 www.ijaem.net ISSN: 2395-5252

recall = j/(i+1e-8)
frscore = 2*precision*recall/(precision+recall)

Then, average of all the frscores is calculated and if the frscore of the current frame is greater than the best frscore then, then this frame would be given the label 1 i.e it is considered as keyframe otherwise 0. In this way, we iterate through each frame to assign labels to each frame.

Semantic segmentation is the major task in video summarization. We build a fully convolutional sequence network for this purpose which is proposed in [12]. The network contains 8 sequential convolutional layers. First 2 convolutional layers contain 2 temporal convolution layers each of which is followed by a batch normalization layer and an activation layer which uses ReLU activation function. Next 3 layers contain 3 temporal convolutions each of which is also followed by a batch normalization layer and a ReLU activation layer. First 5 layers contain a maxpool layer at the end. The 6th and 7th layers contain one temporal convolution followed by a batch normalization layer, a ReLU activation layer and a dropout layer. the 8th convolutional layer does not contain dropout. The 1st deconvolutional layer takes the input from the 8th convolutional layer and 2nd one takes input from the max pool layer of the 4th convolutional layer and also from the 1st deconvolutional layer and gives the final output. The input to the model is the feature vector of the video in which is extracted from GoogleNet model and the output is a binary vector specifying whether each frame is a key frame or not. This binary vector is then converted into a vector of probabilities by passing through a softmax layer. The input to this network is of dimension N x F, and the output is of dimension NxC, where N is the number of frames in the input video and F is the shape of the feature vector of a frame in the video and C is 2 since we find scores for two classes i.e keyframe or not a keyframe.

We use 80/20 train-test split i.e 10 among 50 videos are randomly selected for testing. We train the model with 100 epochs and with a batch size of 8. We use adam optimizer. The loss function used is negative log likelihood loss since it is used in the training of classification problems with a number of classes. Here there are 2 classes i.e. keyframe and not-keyframe. The loss is calculated based on the predicted score from the model and the label given to each frame in the video for each batch.

For evaluating the testing videos, we use the metric fscore which is calculated by the following method.

Calculation of fscore:

The predicted binary vector contains frame scores. The number of keyframes predicted by the model would be greater than 15% of the total number of frames. To reduce the number of frames to 15% of total frames, we find key shots by the following method.

We calculate the mean of predicted scores of frames in a segment.These means of each segment are stored in an array pred_mean. Now, we use the knapsack methodto determine the keyshots. We try to select segments so that the number of frames in the selected segments would sum upto 15% of total number of frames and also maximize the total value of pred_mean. These selected segments are the keyshots. then we label each frame in these keyshots as 1 and others 0.

Now to evaluate the model, the obtained summary y_{pred} is compared with ground truth summary binary vectors. We calculate Precision, recall and fscore user summary and we take the mean of them to all user summaries.For each user or ground truth summary y_{user}

Overlap = sum(y_{user}*y_{pred})
Precision, P = Overlap/sum(y_{pred})
Recall, R = Overlap/sum(y_{user})
fscore = 2*P*R/(P+R)

We evaluate the video summarization model trained on TvSum using the SumMe dataset which is also a standard dataset for video summarization. It contains 25 videos each of which is annotated with 15 human summaries. This SumMe dataset is used in so many studies on video summarizing. In this dataset, each video is of the duration of 5 to 6 minutes.

We create a preprocessed file for this dataset similar to that created for TvSum dataset. And the evaluation metric we use is the fscore which is calculated by the same method used in evaluation of testing videos. The results obtained are shown in Table II.

B. Video Captioning

Given an input output, the aimis to generate a caption for the video using sequence to sequence - video to text[13].

The dataset we use for training this sequence to sequence model is MSVD dataset - Microsoft Video Description Corpus[19], which is a standard dataset prepared for video captioning. This dataset contains around 2000 video snippets from youtube. It contains around 10 to 15 captions for each video. There is no proper dataset available. Due to its large size, for each video, youtube id and the start and end timings (in seconds) are given. So,

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 8 Aug 2021, pp: 1447-1457 www.ijaem.net ISSN: 2395-5252

we download the videos manually from youtube and trim them manually according to the start and end times given in the dataset. The dataset itself comes with a train/test split. But we use all the testing videos for evaluating the model. The training data is split for training and testing in the ratio of 85/15.

We use keras deep learning framework for building, training and validating the model. We build a LSTM model since it has achieved state-of-art results in image captioning. LSTM, abbreviated as Long Short-Term Memory Network is a special type of Recurrent Neural Network which is designed to overcome gradient problems being faced in RNN.

We use Encoder Decoder LSTM which is also called sequence-to-sequence LSTM. Here the encoder network encodes each frame in the video, one at a time and the video is represented as a vector, then the decoder network decodes from that vector to a sequence of words, one word at a time, forming a sentence.

Generally, networks used for the purpose of sequence to sequence modelling makes use of two LSTMs, one for encoder network and the other for decoder network. We use one LSTM both for encoding and decoding so that parameters are shared between encoder and decoder. In our model, two LSTMs are held together. The hidden state representation from the first LSTM is given as input to the second LSTM The representation of the model we use is shown in the figure.

We preprocess the dataset by adding <bos> and <eos> to each caption given for the videos. Then we create vocabulary. We create this by tokenizing the captions with 4 to 10 words in the whole dataset. We include 2000 words in the vocabulary which have maximum frequency. Any caption generated by the model would contain any one of these words.The reason for selecting 2000 words is that the number of unique words in the captions given in the dataset is around 2000. If more than 2000 words are taken, then there may be duplicate words generated in the vocabulary. Also, for each of the videos in the dataset, we create a binary vector (text_sequence) - for each of the words in the vocabulary, if it is present in the captions given to the video, then 1, otherwise 0. For the whole dataset, this binary vector would be of dimension (number of videos, number of words in the vocabulary i.e 2000). For feature extraction, we use a pretrained VGG16 model trained on ImageNet dataset[14], we could use GoogleNet itself for video captioning also(sec 3.1-B), but we just want to explore VGG16 also. VGG16 also has the best accuracy in the classification of images. Input to this model is an image of dimension (224,224,3). We give video as input to the model. Model returns a feature vector for the video.

In the figure 6, the architecture of our video captioning model is shown. Encoder_inputs is nothing but the feature vector of the video. The text_sequence (creation is given in the above paragraph) is the input to the decoder layer (decoder_inputs in figure). The encoder_lstm layer encodes the feature vector of each frame, one at a time and creates a hidden representation. The decoder_lstm receives this hidden representation and adds padding. Once all the frames are encoded, then decoder_lstm is prompted with <bos> after which it starts generating a caption frame by frame as shown in figure.

We train the model with 150 epochs and a batch size of 320. The loss function we use is the categorical cross entropy function since it is used in the problems of classifying among multiple classes. Here the decoder has to classify the words in the vocabulary to be present or not present in the caption. The Loss vs epoch graph and Accuracy vs epoch graph are shown in figures 4 and 5 respectively. The final sentence or the caption is formed using greedy search.

We evaluate our model on the testing data of the MSVD dataset. It contains 92 videos along with 15 to 18 captions for each video. The extracted features of each video is given as input to the model and the caption generated by the model is compared with the captions given by users. We use BLEU score because it is quick to calculate and also language independent. It basically matches the tokens in the reference sentence and the testing sentence one by one and returns the percentage of matched tokens.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 8 Aug 2021, pp: 1447-1457 www.ijaem.net ISSN: 2395-5252

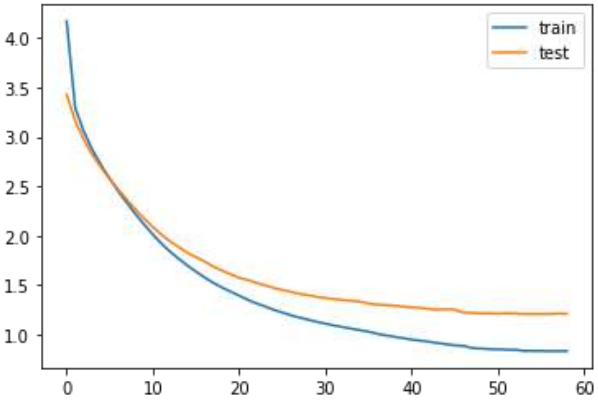

Figure.4. The loss vs epoch graph . X-axis represents the number of epochs , Y-axis represents the loss obtained. Since, the loss remained almost the same from 55th to 60th epoch, training has stopped there as we use early stopping.

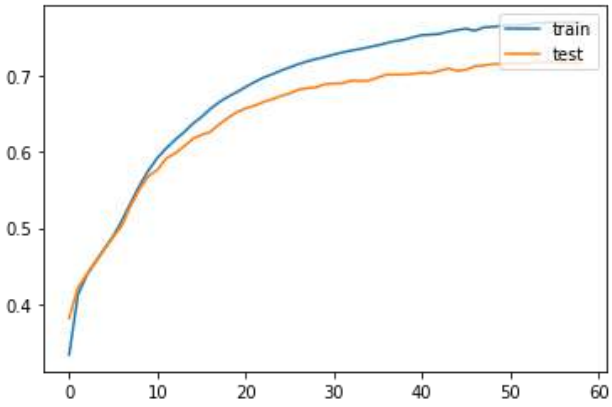

Figure.5. The accuracy vs epoch graph. X-axis represents the number of epochs , Y-axis represents the accuracy obtained.

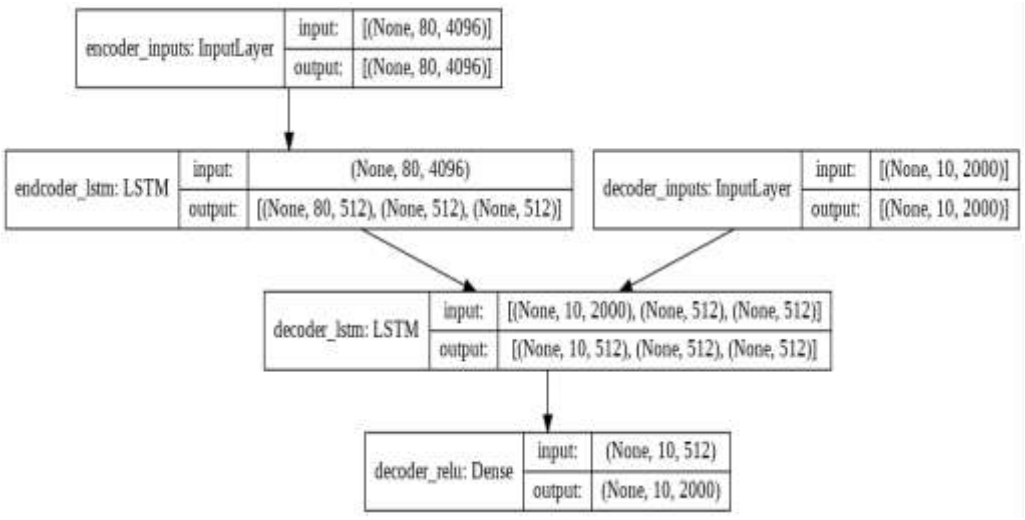

Figure.6. Architecture of our video captioning encoder-decoderLSTM model

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 8 Aug 2021, pp: 1447-1457 www.ijaem.net ISSN: 2395-5252

Example:
Reference sentence = ['the', 'boy', 'is' , 'playing']
test sentence = ['the', 'girl', 'is' , 'playing']
BLEU score = 0.75 means that there is 75% percent match between the sentences. We use the Sentence BLEU score which
is used for evaluating a sentence against multiple sentences as there are 15 to 18 captions given in the dataset.

C. Integration of Video Summarizing and Captioning models

We integrate the models in such a way that the output of the summarization model i.e the summarized video is fed as input to the captioning mode. As a result, the final output is a summarized video along with a caption to that summarized video. These two models can also be used individually depending upon the user's choice.

IV. RESULTS AND ANALYSIS

A.Experimental Results of video Summarizing: In video summarizing, the precision, recall and fscore obtained for testing videos at 100th epoch are shown in Table I.

TABLE I
EVALUATION RESULT OF 100 EPOCH

VideoID	Precision	Recall	Fscore
48	0.401	0.386	0.393
16	0.617	0.609	0.613
13	0.361	0.352	0.357
49	0.499	0.489	0.494
3	0.547	0.540	0.543
21	0.492	0.486	0.489
12	0.573	0.565	0.569
27	0.464	0.460	0.462
42	0.510	0.491	0.500
44	0.602	0.600	0.601
Mean	0.507	0.498	0.502

The mean of the results obtained for SumMe dataset, which we considered for validation is shown in the table below:

TABLE II EVALUATION RESULT OF SUMME DATASET

Dataset	Precision	Recall	fscore
SumMe	0.412	0.381	0.406

As shown in Table I and Table II, the fscores obtained for the TvSum and SumMe datasets respectively, are almost the same as state-of-the-art fscores which are obtained by using LSTMs[3,6]. Also, since our method uses Convolutional networks instead of Recurrent Neural Networks, the computational complexity is less and results obtained are the same.

B. Experimental Results of Video Captioning:The average BLEU score of all the 92 videos which are taken for validation came out to be 0.689. The bleu score obtained for the testing videos is good compared to some of the approaches for video captioning [8,10]. Captions generated for some of the videos by our model are shown in figures 7,8,9,10 and 11. It is observed that our model is able to generate appropraite captions in the case of cooking videos, human activity videos and activitis of only some animals such as cat,dog etc. The reason for this is the lesser number of words in the dataset, thus in the vocabulary.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 8 Aug 2021, pp: 1447-1457 www.ijaem.net ISSN: 2395-5252

Figure.7. Two frames in a video. Caption generated for this video is "woman is stirring rice in the pot" which is appropriate and complete.

Figure.8. Two frames in a video. Caption generated for this video is "Man is performing on the stage which is appropriate and complete but not specific(playing guitar).

Figure.9. Two frames in a video. Caption generated for this video is "Man is holding a small" which is appropriate but not complete.

Figure.10. Two frames in a video. Caption generated for this video is "Cat is playing the piano" which is very appropriate.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 8 Aug 2021, pp: 1447-1457 www.ijaem.net ISSN: 2395-5252

Figure.11. Two frames in a video. Caption generated for this video is "Puppy is playing with a toy" which is inappropriate.

In video captioning, the average BLEU score of all the 92 videos which are taken for validation came out to be 0.689. The bleu score obtained for the testing videos is good compared to some of the approaches for video captioning [8,10].Captions generated for some of the videos by our model are shown in figures 7,8,9,10 and 11. It is observed that our model is able to generate appropriate captions in the case of cooking videos, human activity videos and activities of only some animals such as cat,dog etc. The reason for this is the lesser number of words in the dataset, thus in the vocabulary.

V. CONCLUSION AND FUTURE SCOPE

In this paper, we have proposed fully convolutional sequence networks(FCN)for video summarization and LSTMs for generation of a caption.

For video summarization, we proposed a model that is provoked by FCN in semantic segmentation and adopted semantic segmentation networks. It is found that the model obtains good competitive performance in comparison with other approaches that use Long Short Term Memory. However, our method for video summarization is not bound to Fully convolutional Semantic Network variants and using the same strategies, it is possible to convert almost any segmentation network for summarization of a video.

For Video Captioning, we established descriptions using the Sequence to sequence model, where the frames are read consecutively and then the words are developed consecutively and this permits us to hold input and output of the variable length. Our model considerably gives favor from extra data and recommending that it has a high model ability .

As a future work, we planned to examine more about semantic segmentation models and establish a counterpart model for summarization of a video reducing the processing times by processing more frames per second. Also, for captioning, we would come up with caption generation at each scene change point. We also plan to increase the size of vocabulary by using two or more datasets, as a result of generation of incomplete and inappropriate captions would decrease.

ACKNOWLEDGEMENT

This work would not have been possible without the contribution of Merin Meleet, Assistant Professor – RV College of Engineering and lab in-charge,Dr. Anala M R, Professor – RV College of Engineering, and Smitha G R, Assistant Professor – RV College of Engineering .We are very thankful for all the help they provided throughout the course of this work.

We also like to thank KZhou et al. for making the preprocessed datasets for SumMe and TvSum datasets available easily.

REFERENCES

[1]. Ashenafi Workie;Rajesh Sharma; Yun Koo Chung, "DigitalVideo Summarization Techniques : A Survey"International Journal of Engineering Research and Technology, Jan 2020.

[2]. Aidean Sharghi; Jacob S. Laurel; Boqing Gong,"Query-Focused Video Summarization: Dataset Evaluation, and A Memory Network Based Approach",IEEE Conference on Computer Vision and Pattern Recognition,2018.

[3]. Anurag Sahoo; Vishal Kaushal; Khoshrav Doctor;Suyash Shett; Rishabh Iyer; Ganesh Ramakrishnan,"A Unified Multi-Faceted Video Summarization System", IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[4]. Kaiyang Zhou;Yu Qiao;Tao Xiang,"Deep Reinforcementlearning for Unsupervised video summarization with Diversity-

Representativeness Reward";The Thirty - Second AAAI Conference onArtificial Intelligence(AAAI-18), 2018.

[5]. Varun Luthra; Jayanta Basak; Prof.Santanu Chaudhury; K.A.N. Jyothi, "AVideo Summarization Approach based on Machine Learning**",**International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP), 2008.

[6]. Behrooz Mahasseni; Michael Lam; Sinisa Todorovic."Unsupervised Video Summarization with Adversarial LSTM Networks"; 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[7]. Yujia Zhang;Michael Kampffmeyer;Xiaodan Liang;Min Tan;Eric P. Xing, "Query-Conditioned Three-Player Adversarial Network for Video Summarization", IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018.

[8]. Xin Wang; Wehnu Chen; jiawei Wu, "Video Captioning via Hierarchical Reinforcement Learning", CVPR 2018.

[9]. H. Raksha; G. Namitha and N. Sejal, "Action based Video Summarization", IEEE Region 10 Internation Conference, TENCON 2019.

[10]. Shaoxiang Chen; Ting Yao; Yu-Gang Jiang, "Deep Learning for Video Captioning: A Review ", International Joint Conference on Artificial Intelligence(2019).

[20].

[11]. Christian Szegedy ; Wei Leu; Scott Reed, "Going deeper with Convolutions, IEEE Conference on computer Vision and Pattern Recognition, CVPR 2014.

[12]. Mrigank Rochan ; Linwei Ye; Yang Wang, "Video summarization Using Fully Convolutional Sequence Networks" , European Conference On Computer Vision, 2018.

[13]. Subhashini Venugopalan; Kate Saenko, "Sequence to Sequence -Video to Text", International Conference on Computer Vision, 2018.

[14]. Karen Simonyan ; Andrew Zisserman, "Very Deep Convolutional NetworksFor Large -Scale Image Recognition", CVPR 2016,pp. 1646-1654.

[15]. Silvio Olivasti and Fabio Cuzzolin . "End-to-End Video Captioning",International Conference on Computer Vision Workshop(ICCVW), 2019.

[16]. Christina Newberry, "25 Youtube statistics that may Surprise you" , 'HootSuite', 2 Feb 2021.

[17]. Yalesong, "TVSum: Title-based Video Summarization dataset", CVPR 2015. http://people.csail.mit.edu/yalesong/tvsum.

[18]. Michael Gygli; Helmut Grabner; Hayko Riemenschneider; Luc Van Gool "Creating Summaries from User videos", ECCV 2014. https://gyglim.github.io/me/vsum/index.html

[19]. Zuxwan Hu et al. "Deep Learning for Video Classification and Captioning" Frontiers of Multimedia Research, 2018.