JOURNAL OF NORTHEASTERN UNIVERSITY

Volume 25 Issue 04, 2022 ISSN: 1005-3026 https://dbdxxb.cn/ Original Research Paper

SUMMARIZATION OF CAPTIONS GENERATED FOR A VIDEO THROUGH LSTM AND GRU USING TF-IDF VECTORIZATION

A. Preethi[1], Dr. P. Dhanalakshmi[2]and Dr. T. Thiruvengatanadhan[3]

[1]Research Scholar, Department of Computer Science and Engineering, Annamalai University, Tamilnadu, India

[2]Professor, Department of Computer Science and Engineering, Annamalai University, Tamilnadu, India

[3]Assistant Professor, Department of Computer Science and Engineering, Annamalai University, Tamilnadu, India

Abstract

Video data is very much helpful in easy understanding of a particular event. Nowadays, the video lectures play a major role among the students to have a better understanding of the concepts. Cooking videosin YouTube also found to be more interesting among women. But when these kinds of videos are too lengthy, the people won't watch the video fully. So, if the video is summarized, then the people may have better understanding and save their time in watching the full video. Summarization helps to save the storage space and time. In this work, video shot boundary detection is performed to identify the number of shots in the lengthy video. After that, the captions are generated for the shots of video using video captioning technique. Finally, the dense captions generated are summarized using TF-IDF method. Thus, the long video could be summarized with the help of very few lines of natural language sentences.

Keywords – TF-IDF, Caption, Shots

1. Introduction

A video is an interconnected sequence of frames with respect tospace and time. So, it needs more storage space.A caption is nothing but the natural language sentences generated for the scenes in the video. Instead of watching a full video, the user can easily understand what is the content of the video by reading the captions. Generating captions for all the frames in the video is unnecessary and time consuming. Because all the frames in the video doesn't contain the important information for generating captions. So, video shot boundary detection is necessary to find out the scene change in a video. The shots will contain only the start and end frame of a particular scene by removing the redundant frames in between. With the help of these shots, captions can be generated and then summarized finally using TF-IDF.

2. Related work

Albeer et al., (2022) suggested a method for video summarization using TF-IDF [2], where the keywords are extracted to generate the precise text for YouTube video. The text preprocessing is done by removing all the English stop words and performing stemming. The next step is done for generating unique words from the text by calculating the word count in all documents. The keywords are identified with the help of TF-IDF to generate the summary. The summarized text is evaluated using Recall-Oriented Understudy of Gisting Evaluation (ROUGE).

Submitted: 21/10/2022 **Accepted**: 30/11/2022

Alrumiah et al., (2022) suggested LDA-based summarization [3] method that summarizes text, audio and visual content of educational videos. The given captions for the video are first pre-processed by truncating the punctuations and converting the entire sentence into lowercase. Next an id2word dictionary is modelled that contains the index and mapping for every word called the word corpus. This corpus is sent to the Gensim LDA method that segregates the group of words based on certain topics. The generated final summary of sentences isevaluated against the human translated summary of sentences by ROUGE.

Bendraou et al., (2021) proposed Singular Value Decomposition (SVD) [5] approach to accurately detect the shot boundaries in a video. This method involves two steps., 1) static segment verification and 2) shot transition identification. The video is first spilt into static and dynamic segments. In order to detect whether the segment is static, the Concatenatedblock-based histogram (CBBH) is generated for each frame. For every identified static segment, CBBH features for initial and last frame is calculated. In the second step, Cut Transition (CT) and Gradual Transition (GT) is identified. Once the segment is detected, the distance between two consecutive frames is calculated and it is compared against the threshold. The datasets used here are TRECVid 2001, 2002 and 2005.

Nandini et al., (2022) proposed a method for shot boundary detection where the frames are converted to gray scale and the edge detection techniques such as Robert, Canny and Sobel are applied. The three main steps involved here are feature extraction, abrupt shot detection and keyframe extraction. The features are extracted by Binarized Local Binary Pattern (BELBP) [11] with the help of histogram generation. After that, the Euclidean distance between two frames is computed to find the dissimilarity between the frames where the abrupt cut occurs. After identifying the shots, the keyframes are extracted to get the important information about the video. The keyframes are selected by computing the high coefficient of difference inside a shot.

Gao et al., (2020)proposed an attention-based LSTM model for video captioning. Here, two types of CNN are used as an encoder to extract the features from the given video frames. For every frame, ResNet produce a total of 2048 feature vectors and C3D produce a total of 4096 feature vectors. The features obtained from CNN are concatenated and fed to the hLSTM [8]. The hierarchical LSTM (hLSTM) framework consists of two LSTMs. The first layer is used to decode the obtained features from CNN. The second layer is responsible for generating the captions with the help of previous hidden state and output from the first LSTM layer. In this hLSTM network, both temporal and spatial attention model is incorporated to efficiently generate the captions for the video.

3. Proposed work

The proposed framework for video summarization involves three steps: 1) Video Shot Boundary detection 2) Video captioning and 3) Video summarization using TF-IDF.

3.1 Video shot boundary detection

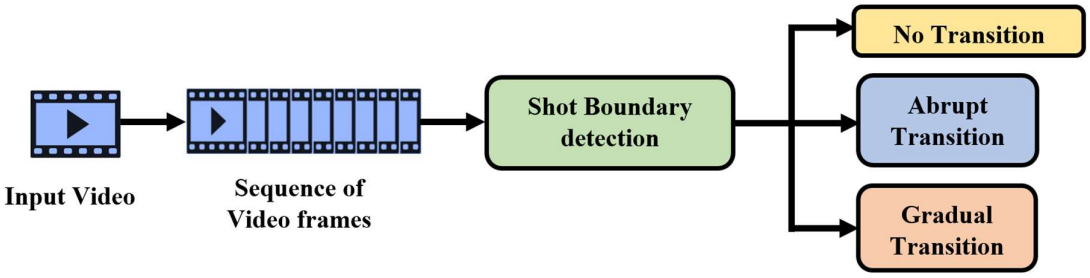


Fig. 1. Shot boundary detection

Video shot boundary detection is the process of identifying the shots inside the video. A shot boundary separates one shot from another. A shot is the continuous set of frames taken using single camera. The shot boundary is classified in two ways: 1) abrupt transition (hard cut) and 2) gradual transition (slow cut). The gradual transition may be further classified into four types such as: wipe, dissolve, fade-in and fade-out as shown in Fig.1. It is identified using both machine learning and deep learning methods. The machine learning methods such as Color histogram, K-means and SVM. The deep learning method used for shot detection is 1 D – CNN.

Color Histogram

Each image frame is divided into blocks and the color histogram of all blocks is computed to get the frame'sfull color histogram. The three channels R, G and B are split into 16 intervals. The distance of every 3*16 (i.e., 48) intervals gives the color histogram of two image frames. The distance value is given by:

$$D(H_i, H_j) = \sum_{r=1}^{16} \sqrt{(h_{ir} - h_{jr})^2} \quad (1)$$

Where$H_i$andH_j indicates the color histogram of the frame *i* and *j*.

h_{ir}andh_{jr}indicates the number of colours in the r^{th}interval of two given frames

When the calculated value of distance among two frames is very small then it is said to consecutive/next frames, or else it is identified as a shot/change as depicted in Fig. 2.

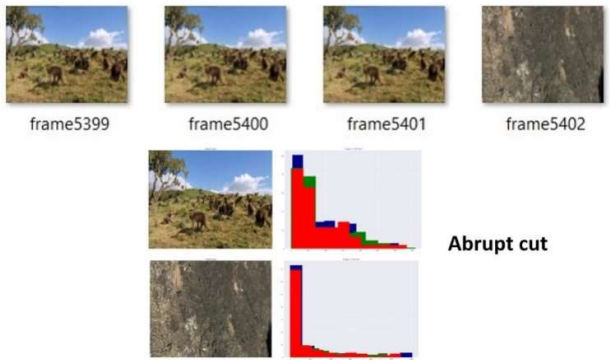


Fig. 2. Abrupt Transition

K-Means

The K-Means clustering will make the video frames fall into different cluster groups. Every sub cluster is identified as the sequence of frames in that cluster itself. For each sub cluster, the mean value of sequence of frame differences (*average_val*) is computed. It is compared against the threshold value Th_b (where $Th_b = \alpha * average\_val$). When the successive differencevalue of frames among sub clusters$f_i \geq Th_b$, then it is split as sub shot. If the value of $f_i \leq Th_b$, then the sub clusters are merged. After that, the mean of newly formed sub clusters is computed. The abrupt cut threshold is taken as $T_a = \beta * average\_val$, if the successive frame difference value $f_i$in the other sub clusters is more than T_a and there is no successive subclusters with atleast 3 image frames,then there is an abrupt cut. When there is successive sub cluster with atleast 3 image frames, it is said to be gradual cut as shown in Fig. 3.

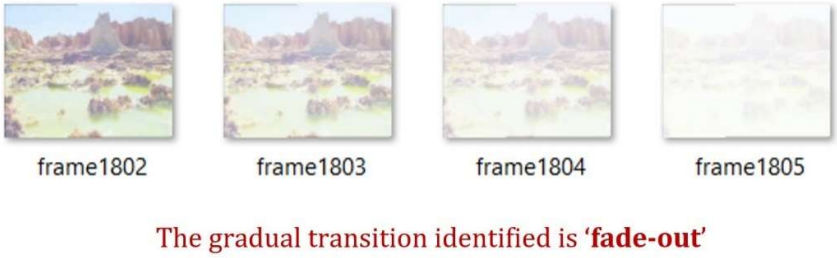


Fig. 3. Gradual transition detected using K-Means

SVM

The multi-class Support Vector Machine identifies the optimal hyperplane which is able to effectively separate the given image frames into three categories, namely; no transition (Class 1), abrupt (Class 2) and gradual transition (Class 3) as shown in Fig. 4. The Radial Basis Function (RBF) kernel is used here. Here "one-against-one" SVM is used to find the shot boundary between frames. So, there are *k(k-1)/2* binary classifiers and it is trained. Finally, there are 3 binary classifiers with class pairs (1,2), (1,3) and (2,3). The strategy of voting is adopted to find the output with a greater number of votes as given in Fig. 5.

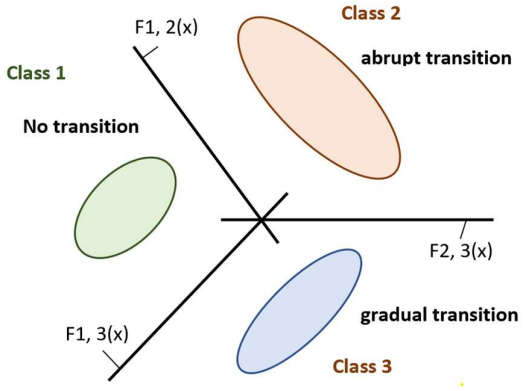
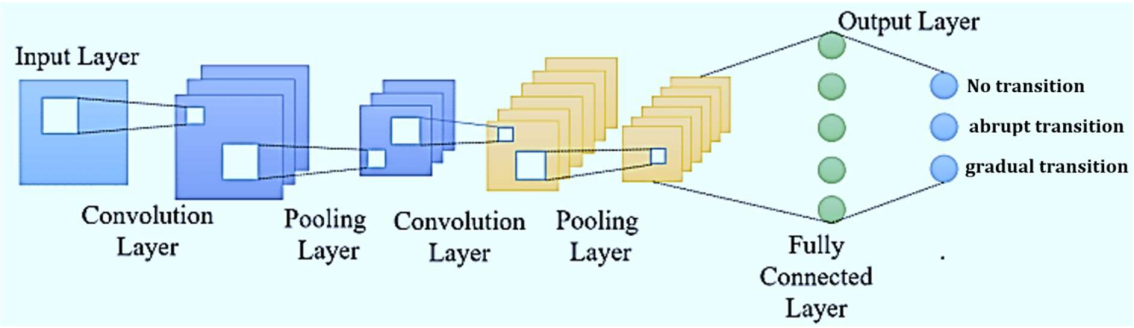


Fig. 4. SVM for Multiclass classification

Fig. 5. Gradual transition detected using SVM

1D-CNN

1D – CNN is a deep neural network architecture that includes convolution, activation and pooling with final softmax layer for classification. Here the input layer is taken as 1 X 36 and



there are 2 convolution layers and 2 pooling layers with filter of size 1 X 3, stride is 1 with no padding. Here, ReLu is applied as an activation function. The final softmax layer gives the three-class output as, no transition, abrupt transition and gradual transition as shown in Fig. 6.

Fig. 6. 1D-CNN for shot detection

3.2 Video Captioning

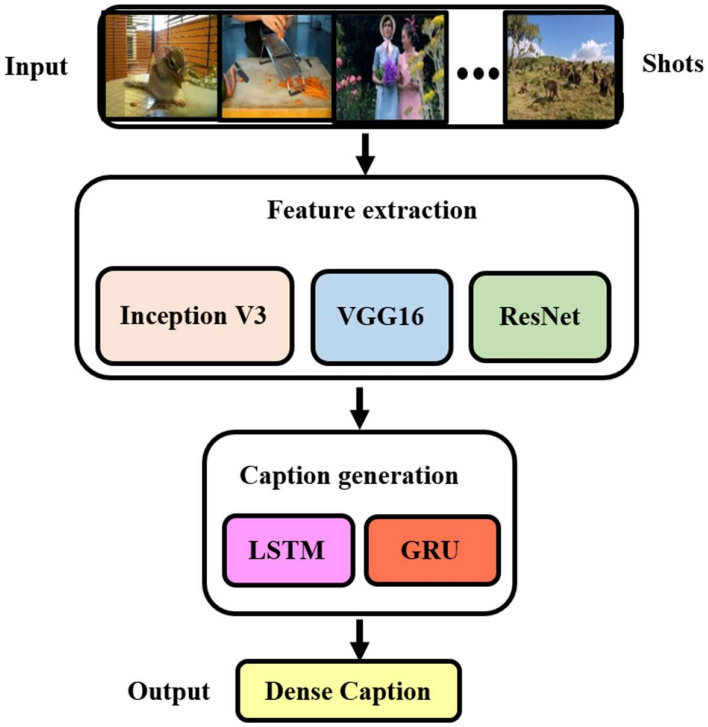


Fig. 7. Video captioning model

Video captioning is the process of generating the appropriate text information for the corresponding scenes in a video. Here, this captioning is achieved through encoder decoder framework model. The feature extraction part is done in encoder and caption generation part is done in decoder as shown in Fig. 7.After identifying the shots, feature extraction is done through pre-trained networks such as InceptionV3, VGG16 and ResNet. The extracted features are fed as an input to the LSTM and GRU. These recurrent neural networks generate the captions with the help of word embeddings. Finally, the dense caption explaining the content of video is obtained.

InceptionV3

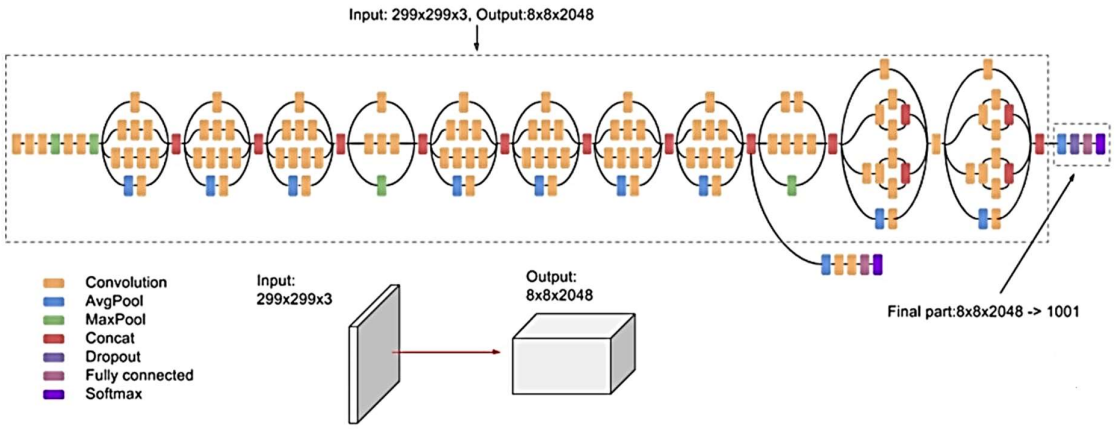


Fig. 8. Structure of InceptionV3

InceptionV3 is a pretrained CNN on ImageNet dataset with an input size of 299 X 299. The detected shots using VSBD is resized and fed as an input to InceptionV3. The top layer used for classification is removed and the fully connected layer is extracted as output. So, a total of 2048 length vector is obtained for each image. During training, a total of 100 epochs is carried out with Adam optimizer using the learning rate of $2e^{-5}$. The total number of layers in InceptionV3 is 48 with 6.4 million trainable parameters.

VGG16

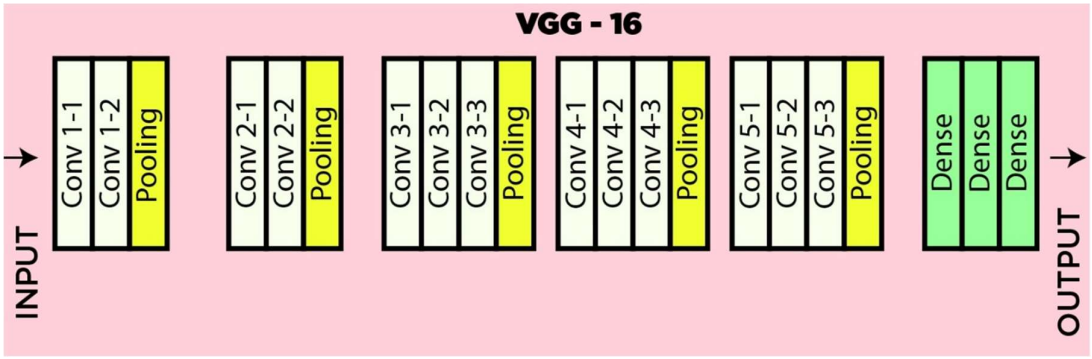


Fig. 9. Structure of VGG16

VGG16 is a pretrained model and it is imported from tensorflow. It accepts the input size of 224 X 224. Here, there are about 13 convolution layers with 3*3 kernel filters. The '*include_top = False*', python code paves the way for feature extraction and removes the classification part. Here a total of 4096 feature vector is obtained from the last fully connected layer for each input image. During training, the total number of trainable parameters here is about 134,268,738 and it takes the training time of 3 hours 40 minutes. There are about 100 epochs with a batch size of 20.

ResNet– 152

ResNet – 152 is a pretrained model that make use of identity function and skip connection. The input size of the image frame is resized here as 112 X 112. The convolution operation is carried in residual blocks which follows skip connection thus makes it easier for computation. The total number of features extracted here is 2048 as shown in Fig. 10. This has 60.3 million trainable parameters.

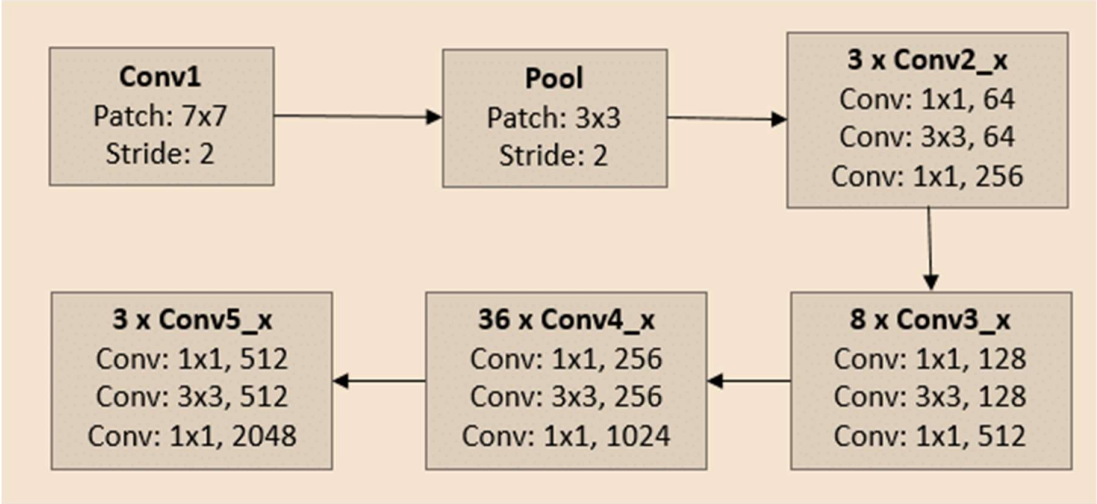


Fig. 10. Structure of ResNet-152

LSTM

Long Short-Term Memory network is mainly used in case of sequential and temporal related connection between the nodes. It can able to store the previous stage information in memory cell and it can be used for prediction of upcoming states. It has 3 gates namely: 1) forget gate 2) input gate and 3) output gate. These gates will modify the information in memory cell. In this caption generation model, LSTM [6] will try to find out the next word with the help of previously encountered words. The entire process of prediction will stop until it encounters the end token as shown in Fig. 11.

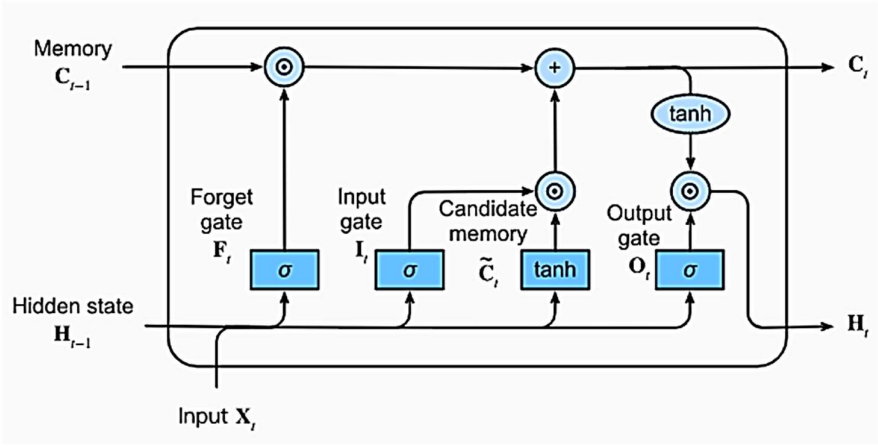


Fig. 11. Structure of LSTM

GRU

Gated Recurrent Unit make use of only 2 gates namely: 1) update gate and 2) reset gate. These two gates are responsible for which information is needed for output prediction and which information should be truncated. So, it reduces the overhead compared to LSTM as shown in Fig. 12.

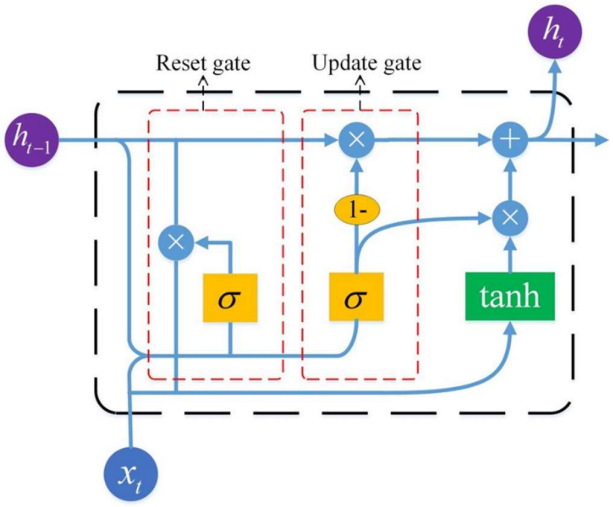


Fig. 12. Structure of GRU

3.3 Video Summarization using TF-IDF

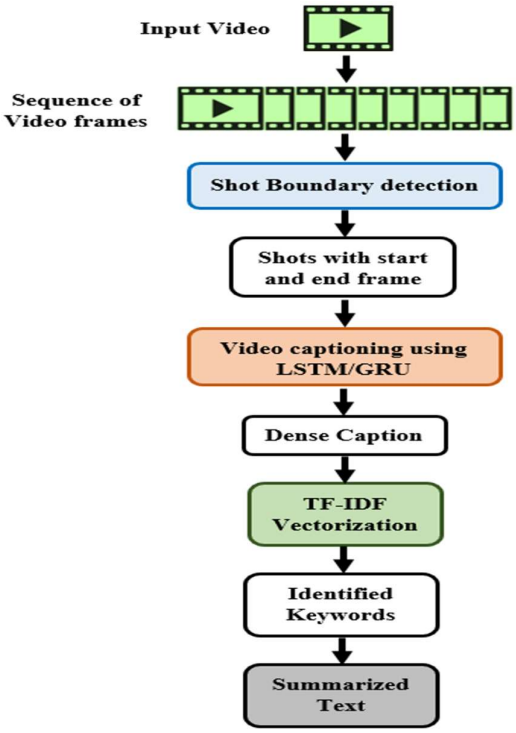


Fig. 13. Flowchart forVideo summarization

The input video is partitioned into small chunks of video with the help of Shot boundary detection method. The detected shots are then sent to LSTM/GRU to generate the dense captions. The dense captions are summarized using TF-IDF vectorization to get a short explanation of the content in the video. In the first step, preprocessing is carried out by removing stop words and then lemmatization of similar words. Then the keyword is identified based on the highest TF-IDF score value. After identifying the keywords, the summarized text based on the keywords is generated with the help of WordNet (lexical database). The overall block diagram of the proposed work is given in Fig. 13.

The generated captions for the video are given as a text document input to the TF-IDF vectorizer. It calculates the score of the frequent term in all documents and return the most repeated keywords. With the help of these keywords, the WordNet lexical database is used to generate the summarized captions for the video.

Term Frequency

It is the value of the count of total times a particular term occurring in the document. This is represented in the form of matrix where the rows indicate total number of documents and the columns indicate the total number of unique terms in all documents as shown in Fig. 14.

$$tf(t,d) = \frac{count\ of\ t\ in\ d}{number\ of\ words\ in\ d} \quad (2)$$

Where, '*t*' indicates the term and '*d*' indicates the document.

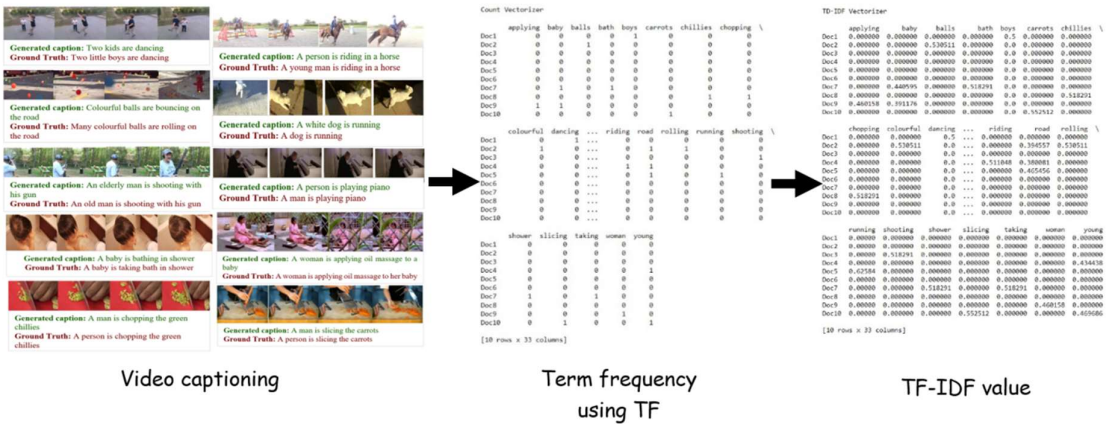


Fig. 14. Evaluation of term frequency using TF-IDF

Document Frequency (DF)

It calculates the total number of documents containing the particular term.

$$df(t) = document\ frequency\ of\ a\ term\ t \quad (3)$$

Inverse Document Frequency (IDF)

IDF calculates theweight of the term in the documents. When the value of *df(t)* increases, it will automatically reduce the value of *idf(t).* When the value of *df(t)* equals *N* (i.e., the term is present in all documents), then *idf(t)* is zero[7].

$$idf(t) = \log(\frac{N}{df(t)}) \quad (4)$$

Now TF-IDF weight of a term can be calculated using the formula,

$$tf - idf(t,d) = tf(t,d) * idf(t) \quad (5)$$

Table. 1.Frequent terms identified using TF-IDF

Document	Identified Terms
Doc1	*boys, dancing*
Doc2	*colorfull, balls, rolling, road*
Doc3	*man, shooting*
Doc4	*young, man, riding, road*
Doc5	*dog, running*
Doc6	*man, playing, piano*
Doc7	*baby, bath, taking, shower*
Doc8	*man, chopping, chillies*
Doc9	*women, applying, bath, baby*
Doc10	*man, slicing, carrots*

The keywords are identified by calculating the TF-IDF value for all the terms in the document as shown in Table. 1.

WordNet

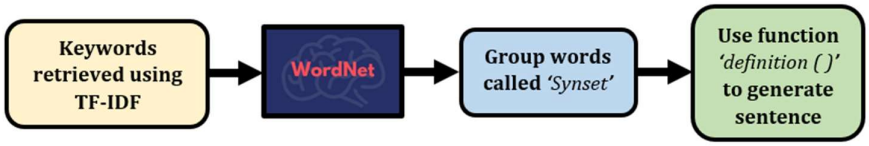


Fig. 15. WordNet for summarization

WordNet is a database which acts as a dictionary for English words. It can be easily used in python as a package from Natural Language Tool Kit (NLTK). WordNet gets the input as keywords returned from TF-IDF and puts the same set of words into a cluster called ***Synset***. After identifying the *Synset*, the function called ***definition()*** is used to make the sentence for the corresponding words in *Synset*.Thus, the summarized text is generated for the video as

```
Boys dancing. Colorful balls rolling on road. The man is shooting.
The young man is riding. A baby taking  shower bath. A man chopping chillies.
A woman applying oil bath to baby. A man slicing carrots.
```

shown in Fig. 15 and 16.

Fig. 16. Summarized text using WordNet

4. Evaluation metrics

In this work, the summarized text is evaluated using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) that is particularly developed for assessing machine as well as automatic text summarization in NLP. ROUGE-1, ROUGE-2 and ROUGE-L for the generated summary is calculated. ROUGE-1 gives the 1-unigram overlap of words in the generated text. ROUGE-2 gives the 2-unigrams overlap in the generated text. ROUGE-L refers to the longest matching

sequence of words in the generated text. The calculated values of ROUGE for the summarized text using TF-IDF on different video data is shown in Table. 2.

Table. 2. ROUGE-1, ROUGE-2 and ROUGE-L for the Test data

Summarized Text using TF-IDF	ROUGE - 1			ROUGE - 2			ROUGE - L		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
V1	0.721	0.793	0.755	0.695	0.766	0.728	0.708	0.785	0.744
V2	0.699	0.782	0.738	0.758	0.811	0.783	0.759	0.744	0.751
V3	0.719	0.759	0.738	0.739	0.798	0.767	0.760	0.789	0.77
V4	0.730	0.791	0.759	0.690	0.738	0.713	0.771	0.729	0.749
V5	0.763	0.801	0.781	0.755	0.718	0.736	0.768	0.780	0.77

The comparison of calculated ROUGE-1, ROUGE-2 and ROUGE-L values for the summarized text is shown in the graph as Fig. 17 below.

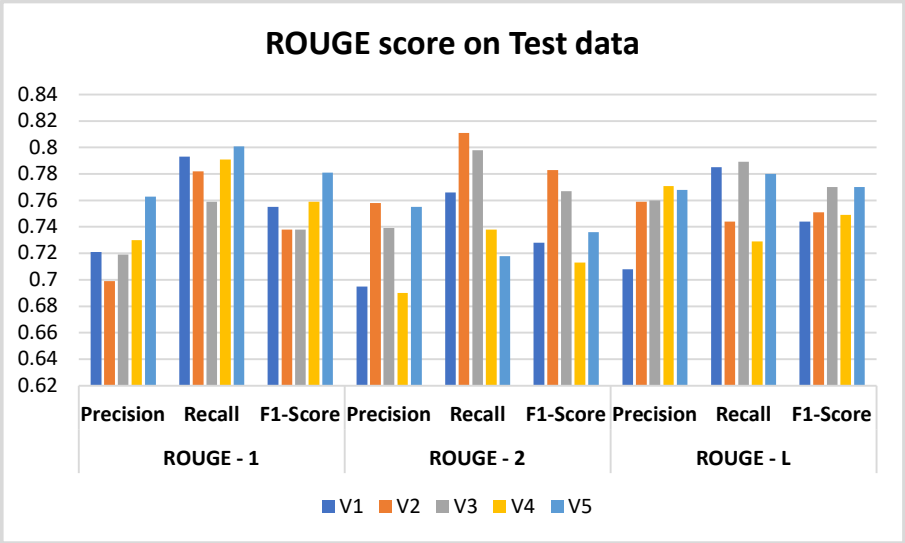


Fig. 17. Comparison of ROUGE values on Test Video data

Conclusion

Thus, in this work, a video summarization model is created which can easily translate a given video with the help of very few lines of sentences. The model first splits the video into smaller sub units as 'shots' with the help of shot boundary detection technique. Then the captions are created for the shots with the help of video captioning technique. The generated dense captions

for all the shots are then summarized into few sentences with the help of TF-IDF method and WordNet.

References

1. Alamuru, S., & Jain, S. (2021). Video event classification using KNN classifier with hybrid features. *Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2021.03.154
2. Albeer, R. A., Al-Shahad, H. F., Aleqabie, H. J., & Al-Shakarchy, N. D. (2022). Automatic summarization of YouTube video transcription text using term frequency-inverse document frequency. *Indonesian Journal of Electrical Engineering and Computer Science*, *26*(3), 1512–1519. https://doi.org/10.11591/ijeecs.v26.i3.pp1512-1519
3. Alrumiah, S. S., & Al-Shargabi, A. A. (2022). Educational videos subtitles' summarization using latent dirichlet allocation and length enhancement. *Computers, Materials and Continua*, *70*(3), 6205–6221. https://doi.org/10.32604/cmc.2022.021780
4. Apostolidis, E., Adamantidou, E., Metsai, A. I., Mezaris, V., & Patras, I. (2021). *Video Summarization Using Deep Neural Networks: A Survey*. http://arxiv.org/abs/2101.06072
5. Bendraou, Y., Essannouni, F., Aboutajdine, D., & Salam, A. (2017). Shot boundary detection via adaptive low rank and svd-updating. *Computer Vision and Image Understanding*, *161*, 20–28. https://doi.org/10.1016/j.cviu.2017.06.003
6. Bin, Y., Yang, Y., Shen, F., Xie, N., Shen, H. T., & Li, X. (2019). Describing video with attention-based bidirectional LSTM. *IEEE Transactions on Cybernetics*, *49*(7), 2631–2641. https://doi.org/10.1109/TCYB.2018.2831447
7. Dilawari, A., & Khan, M. U. G. (2019). ASoVS: Abstractive Summarization of Video Sequences. *IEEE Access*, *7*, 29253–29263. https://doi.org/10.1109/ACCESS.2019.2902507
8. Gao, L., Li, X., Song, J., & Shen, H. T. (2020). Hierarchical LSTMs with Adaptive Attention for Visual Captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(5), 1112–1131. https://doi.org/10.1109/TPAMI.2019.2894139
9. Liu, A. A., Qiu, Y., Wong, Y., Su, Y. T., & Kankanhalli, M. (2018). A Fine-Grained Spatial-Temporal Attention Model for Video Captioning. *IEEE Access*, *6*, 68463–68471. https://doi.org/10.1109/ACCESS.2018.2879642
10. Mohamed Mansoor Roomi, S., & Maragatham, G. (n.d.). *Video Summarization using Hierarchical Shot Boundary Detection Approach*.
11. Nandini, H. M., Chethan, H. K., & Rashmi, B. S. (2022). Shot based keyframe extraction using edge-LBP approach. *Journal of King Saud University - Computer and Information Sciences*, *34*(7), 4537–4545. https://doi.org/10.1016/j.jksuci.2020.10.031
12. Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J., & Yokoya, N. (2016). *Video Summarization using Deep Semantic Features*. http://arxiv.org/abs/1609.08758

13. Sasithradevi, A., & Mohamed Mansoor Roomi, S. (2020). A new pyramidal opponent color-shape model based video shot boundary detection. *Journal of Visual Communication and Image Representation*, *67*. https://doi.org/10.1016/j.jvcir.2020.102754
14. Song, J., Guo, Y., Gao, L., Li, X., Hanjalic, A., & Shen, H. T. (2019). From Deterministic to Generative: Multimodal Stochastic RNNs for Video Captioning. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(10), 3047–3058. https://doi.org/10.1109/TNNLS.2018.2851077
15. Subudhi, B. N., Veerakumar, T., Esakkirajan, S., & Chaudhury, S. (2020). Automatic lecture video skimming using shot categorization and contrast based features. *Expert Systems with Applications*, *149*. https://doi.org/10.1016/j.eswa.2020.113341