KSUMM: A Compressed Domain Technique for Video Summarization Using Partial Decoding of Videos

Madhushree Basavarajaiah(✉) and Priyanka Sharma

Department of Computer Engineering, Institute of Technology,
Nirma University, Ahmedabad, India
{16ftvphde13, priyanka.sharma}@nirmauni.ac.in

Abstract. Generally, the videos are encoded before storing or transmitting. Traditional video processing techniques are compute intensive as they require decoding of the video before processing it. The compressed domain processing of video is an alternative approach where computational overhead is less because a partial decoding is sufficient for many applications. This paper proposes a video summarization technique, KSUMM, that works in the compressed domain. Based on the features extracted from just the I-frames of the video, frames are classified into a predefined number of classes using K-means clustering. Then, the frame which is located at the border of a class in the sequential order is selected to be included in the summary. The length of the summary video can be customized by varying the number of classes during clustering. The quality of the summary was evaluated using Mean Opinion Scores method and the result shows a good Quality of Experience.

Keywords: Video summarization · Machine learning · Video abstraction
Compressed video processing

1 Introduction

As per the Cisco Visual Networking Index, 75% of the mobile data traffic worldwide will be videos by 2020. Videos are an integral part of many applications. Mostly, a video is compressed after it is recorded for storing or transmission purpose. Even though modern video encoders are very efficient in compressing the video, they take longer time in decoding the video. Traditional video summarization techniques decode the video into a sequence of frames before processing it, which is an additional overhead in summarization process. On the other hand, we have compressed domain video summarization techniques which involve only a partial decoding of the videos in order to perform summarization task. This saves time and space in the overall process of summarization [1–4].

The process of summarization of videos can be defined as the generation of synopsis of a lengthy video so that it can be interpreted in less time. There are two types of video summarization [1]. They are: (i) Static Storyboard and (ii) Dynamic Video Skimming. Story board is a collection of keyframes selected from a static video.

A. K. Luhach et al. (Eds.): ICAICR 2018, CCIS 955, pp. 241–252, 2019.
https://doi.org/10.1007/978-981-13-3140-4_22

Keyframes are the crucial frames of the original video which are included in the summary. Video skimming is the process of selecting important video clips from the original video.

Based on the working domain of the video, the process of video summarization is divided into two categories namely Pixel/uncompressed domain video summarization and Compressed domain video summarization. In uncompressed domain summarization, video is decoded before using it at the pixel level. In compressed domain, video is not decoded completely into frames. But, the information regarding the content of the video is extracted from partially decoded video [2]. Uncompressed domain video summarization involves three main issues as picturized in Fig. 1. The process of compressed domain summarization does not require complete decoding of the video before processing. A partial decoding of only I-frames is sufficient for extracting features from video based on which the key frames can be selected. This process is shown in Fig. 2.

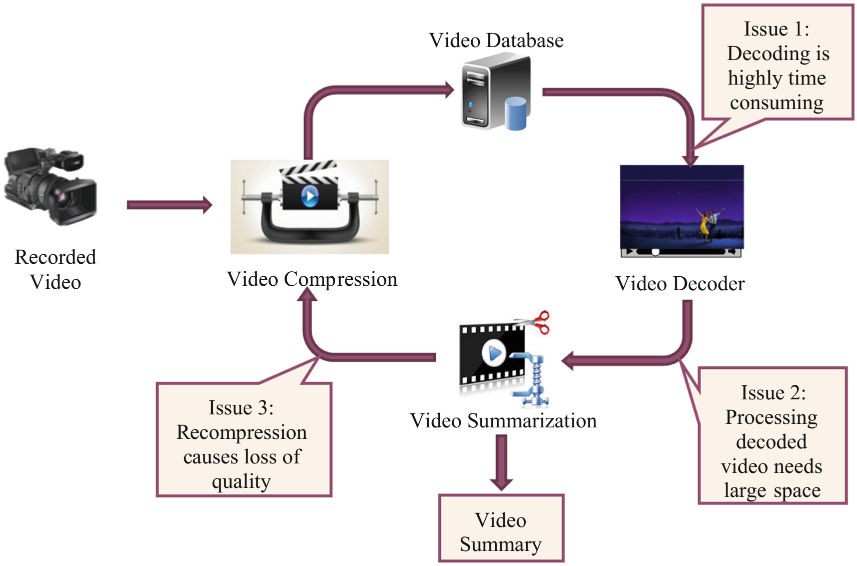
Video Database
Issue 1: Decoding is highly time consuming
Recorded Video
Video Compression
Video Decoder
Video Summarization
Issue 3: Recompression causes loss of quality
Issue 2: Processing decoded video needs large space
Video Summary

Fig. 1. The process of uncompressed domain video summarization and the issues involved in the process.

In this paper, the challenges faced in performing video summarization in the uncompressed video are addressed by implementing a new approach to obtain the summaries of lengthy videos in the compressed domain. The proposed approach, KSUMM, uses machine learning technique to find the frames in a video which are at the major changes in the content of the video. These frames are called keyframes and they are selected for including in the video summary. Video summaries are generated when a user requests for a summary and the quality of the summary is evaluated using a subjective measurement of Human Visual System [HSV]. In this evaluation method,

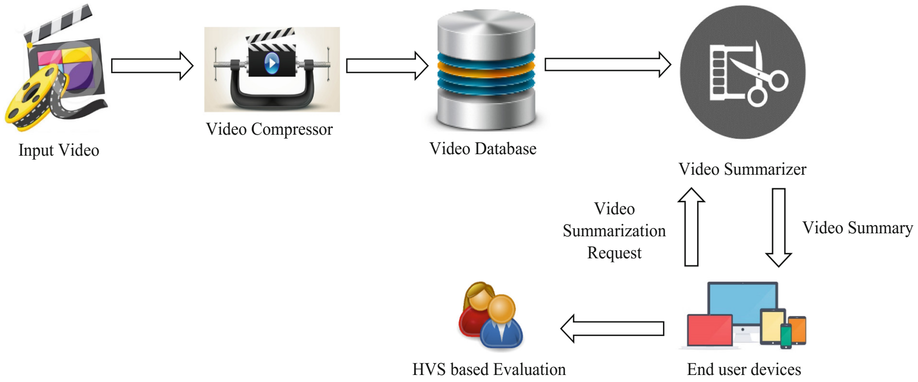

Fig. 2. The process of compressed domain video summarization

the video summary is shown to different users and the opinion score is recorded. A Mean Opinion Score (MOS) is calculated by taking average of all these scores.

The remaining part of the paper is structured as follows. Section 2 gives the literature review on compressed domain video summarization techniques. Section 3 explains the proposed work and the system overview. Implementation details are given in Sect. 4 and results are discussed in Sect. 5 with a trailing conclusion and future work in Sect. 6.

2 Literature Review

Video summarization is traditionally carried out on the uncompressed videos. But, there is a wide scope for research in processing a compressed video as it is less time consuming compared to pixel domain processing. Video summarization is one such problem in which compressed domain techniques are more in demand. In this section, we present a review of literature which use compressed domain techniques for videos summarization.

Video summarization can be performed on the basis of many characteristics of the video. Researchers have tried to implement summarization algorithms using such features of video. Also, summarization is performed in various application domains for example, sports, news, medical and online videos etc. Along with the visual features of the video content, audio and metadata accompanying with the video can also be used for summarization purpose. Similar amalgamation of video and audio features was used to summarize soccer videos by Kiani et al. [3]. A progressive generation mannered method was used to generate video summaries based on the visual content information extracted from the compressed domain video features by Almeida et al. [4, 5]. Yu et al. used semantic features along with metadata available in the compressed video to summarize MPEG [Moving Picture Experts Group] encoded videos [6]. User interaction can also be incorporated in the process of video summarization [7].

Some researchers have worked on specifically H.264/AVC [Advanced Video Coding] encoded videos. A video segmentation method which uses features attained from the entropy decoding of the video is implemented by Schöffmann et al. [8]. Herranz et al. united the content adaptation method into video summarization module [9]. Object tubes extraction was used by Zhong Rui et al. for summarization [10]. Apart from all these methods, compressed domain features like Discrete Cosine Transforms and Motion vectors are also used in summarization process. Video summarization through energy minimization in compressed domain was proposed using graph cut algorithm [10]. With all the literature reviewed in compressed domain, we can say that processing a video in compressed domain gives better results compared to pixel domain video processing [13].

3 Proposed Approach

The proposed work aims at solving the challenges faced in carrying out video summarization in uncompressed domain. This system can offer a video summary when a user requests for it based on the user requirements. Users can have the choice of deciding on the length of the summary. The major steps followed in the implementation are listed below.

Step 1: Extraction of compressed domain features from the video by decoding only the I-frames of the compressed video.
Step 2: Clustering of the frames depending on the features extracted using unsupervised machine learning technique.
Step 3: Selecting the frames which represents a scene change. These frames would later be included in the video summary.

3.1 Proposed Approach

In this proposed work, a summary of a video is created by selecting the important frames from a video sequence. A good video summary storyboard will provide the maximum information present in the input video in as less number of frames as possible. The outline of the proposed work is given away in Fig. 3.

The keyframes obtained from the video give the user a gist of the visual content of the video. In our approach, keyframes are selected when there are changes in the visual content or the scene change.

While compressing a video, MPEG encoder converts it into a sequence of I, B and P-frames. These frames are grouped together in the form of Group of Pictures (GoP). I-frames are the intra - coded frames which appear at the beginning of every GoP, followed by a specific number B and P-frames. This series of frames repeat till the end of the video. The number of these frames is decided by the encoding software [2]. I-frames hold most of the visual data of the video. Hence, only I-frames of the video are decoded for the summarization process here as shown in Fig. 4. The structure of a digital video compressed in MPEG format with I, B, and P-frames is shown in Fig. 5.

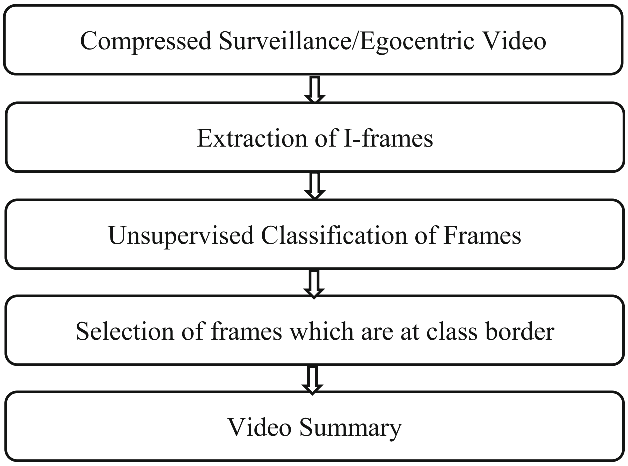
Compressed Surveillance/Egocentric Video
Extraction of I-frames
Unsupervised Classification of Frames
Selection of frames which are at class border
Video Summary

Fig. 3. Flow of the proposed approach

Fig. 4. Sample of decoded I-frames of Basketball 1920 × 1080_50.mp4 video

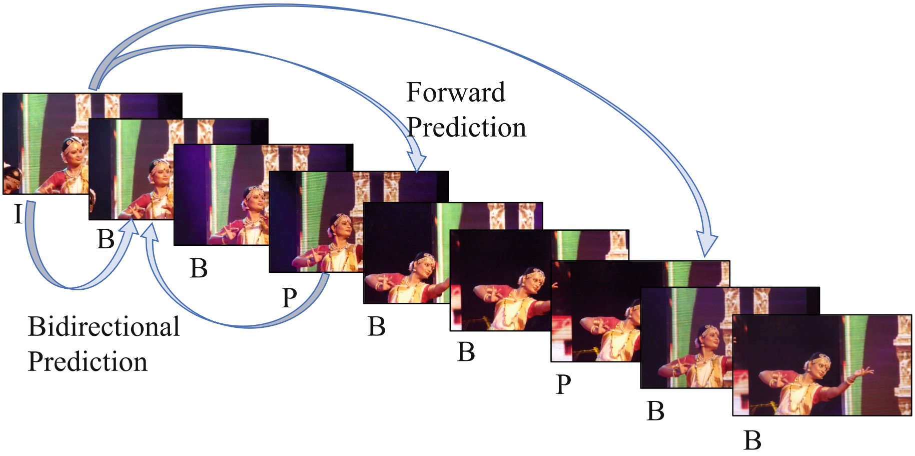

Fig. 5. Frames in a compressed video

DCT coefficients are the important compressed domain features which can be obtained without decoding the video. Because they are calculated while encoding the video to represent the video by removing the high frequency elements in the visual content. Figure 6 shows the DCT images of corresponding I-frames shown in Fig. 4.

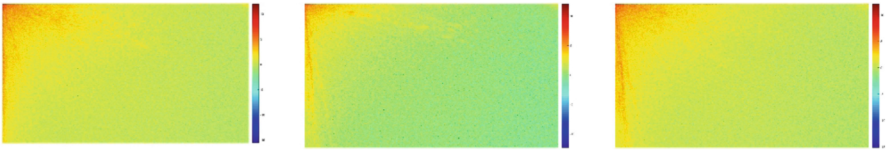

Fig. 6. Corresponding DCT coefficients of I-frames shown in Fig. 4.

Similarly, Motion Vector, an important feature which represents the position of a macroblock with respect to a reference frame, can be extracted from the compressed video without decoding the video completely. Motion Vector can be used to find the movement of objects in the visual content of a video. Motion Vectors calculated from Basketball 1920 × 1080_50.mp4 video are shown in Fig. 7.

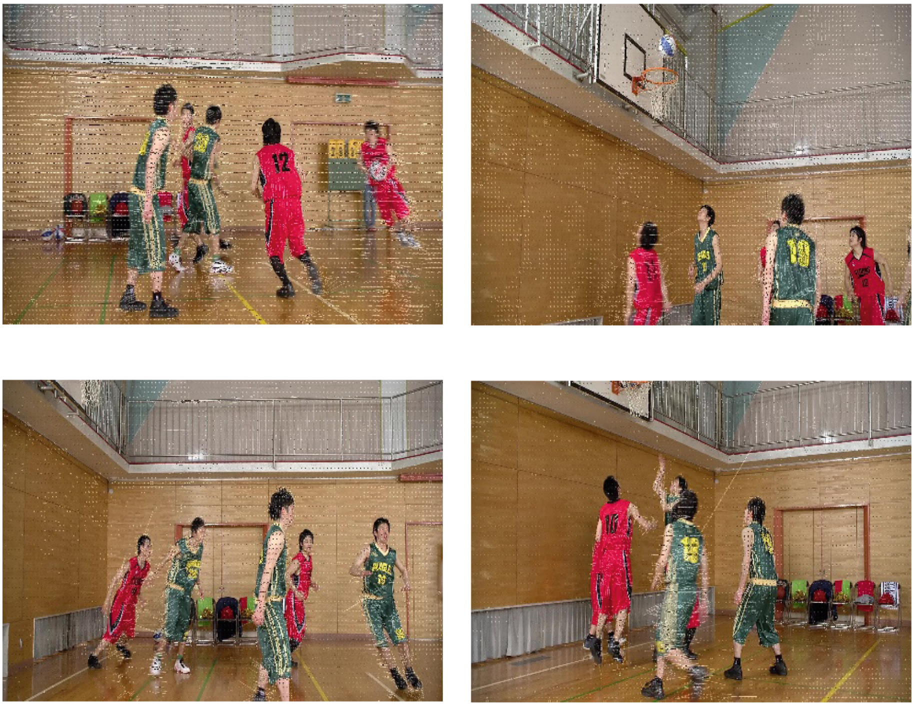

Fig. 7. Motion vectors represented in the frames of a sample video

4 Implementation

A simple and efficient technique is designed to produce a summary of a compressed surveillance/egocentric video. Consider a typical video V, as a set of frames, where f_i is the constituent frame of the video at position i and n represents the total number of frames of the video V.

$$V = \{f_1, f_2, f_3, \ldots, f_n\} \tag{1}$$

Similarly, a compressed video, CV, is a sequence of fi, fb and fp which represent the I, B, and P-frames respectively. A sample of I, B, and P-frames sequence is generalized in Eq. 2. An example sequence for the Eq. (2) looks like IBBPBBPI.

$$CV = \left\{fi_1, fb_1, \ldots, fb_{nb}, fp_1, fb_{nb+1}, \ldots, fb_{2nb}, fp_2, fb_{2nb+1}, \ldots, fb_{3nb}, fi_2, \ldots, fp_{np}, fi_{ni}\right\} \tag{2}$$

The number of I-frames is represented by ni, np represents the number of P-frames and nb is the number of B-frames which appear in sequence between an I and a P-frame. This number can be varied during the encoding of a video.

Only the I-frames of the compressed video are decoded since they contain the maximum information of the video. So, only a partial decoding of the encoded video is performed in the first step. The sequence of I-frames, represented as set I, acts as the input to the second step. The number of I-frames extracted from the video is n.

$$I = \{fi_1, fi_2, fi_3, \ldots, fi_n\} \tag{3}$$

Later, the feature extraction step is carried out to get the scene change or changes in the visual content of the video by extracting useful features from the frames of the video. The possible options for the features are DCT coefficients, Motion Vectors, Quantization Parameter, Color changes in the frames etc. In this implementation, each frame is divided into nine equal parts and the features are extracted based on the color histogram and the matrix having the positions of the greatest color histogram is constructed. This matrix with numerical values extracted from every frame forms the feature set using which the frame clustering is performed in the third step.

Unsupervised classification technique, the k-Means clustering method is used to classify the frames because the k-Means algorithm is well suited for general purpose clustering of numerical data with even cluster size and less number of clusters. The features extracted from the frames form matrix with 27 columns stand for the attributes of the data and the rows of the matrix represent the frames of the video. The feature set is given as an input to the unsupervised classification module that classifies the frames into a predefined set of classes of frames, C_k. k is the number of classes which can be defined by the user. fi_p is the first frame in the class and fi_q is the last frame.

$$C_k = \{fi_p, \ldots, fi_q\} \tag{4}$$

The last step is the selection of keyframes to form the video summary storyboard. The class labels assigned to each frame are sequentially scanned and a frame is selected if there is a change in the class label. For example, fi_1, fi_2 and fi_3 belong to C_1, then fi_1 and fi_4 are selected because fi_4 belongs to C_2. This process continues till all frames in the list are traversed. Finally, all the selected frames put together form the summary of a video, SV. fi_{C1} is the first frame from the class C_1, fi_{C2} is the first frame from class C_2 and so on. fi_{Ck} is the frame at the first frame from the last class C_k. fi_n is the last frame in the series of I-frames.

$$SV = \{fi_{C1}, fi_{C2}, fi_{C3}, \ldots, fi_{Ck}, fi_n\} \tag{5}$$

5 Result Analysis

The implementation of the proposed technique was carried out using Python. MPEG-4 videos were used for testing the summarization method. The number of frames included in the summary depends on the number of scene changes in the video as the technique selects frames based on the content change in the video. For example, a video from VSUMM dataset [12] containing 358 I-frames was summarized to 28 frames using KSUMM. That means a 7.8% of the frames were selected to be a part of the summary when a 5-class unsupervised classifier was used.

Increasing the number of classes increases the number of keyframes selected for the summary and vice-versa. Average complexity of the k-Means clustering algorithm is O (knT), where k is the number of classes, n represents the number of samples and T gives the number of iteration [11]. The summary frames of a sample surveillance video are shown in Fig. 8.

The evaluation of the summarization quality is a difficult process since there are no well-defined measures for the goodness of a summary and the usefulness of the summary majorly depends on the requirement of the user. Hence, the summary along with the original input video was shown to 20 volunteers who were asked to rate the summary quality based on the Quality of Experience (QoE) on a scale of 1 to 5 where 1 being the least and 5 being the best score. Mean Opinion Score (MOS) were calculated for every video summary based on the user rankings. An average score of 4.1 was given to the result shown in Fig. 8. The MOS for different test videos as obtained by the experiment are shown in Table 1. The summary frames of another sample surveillance video from VIRAT dataset [14] are shown in Fig. 9.

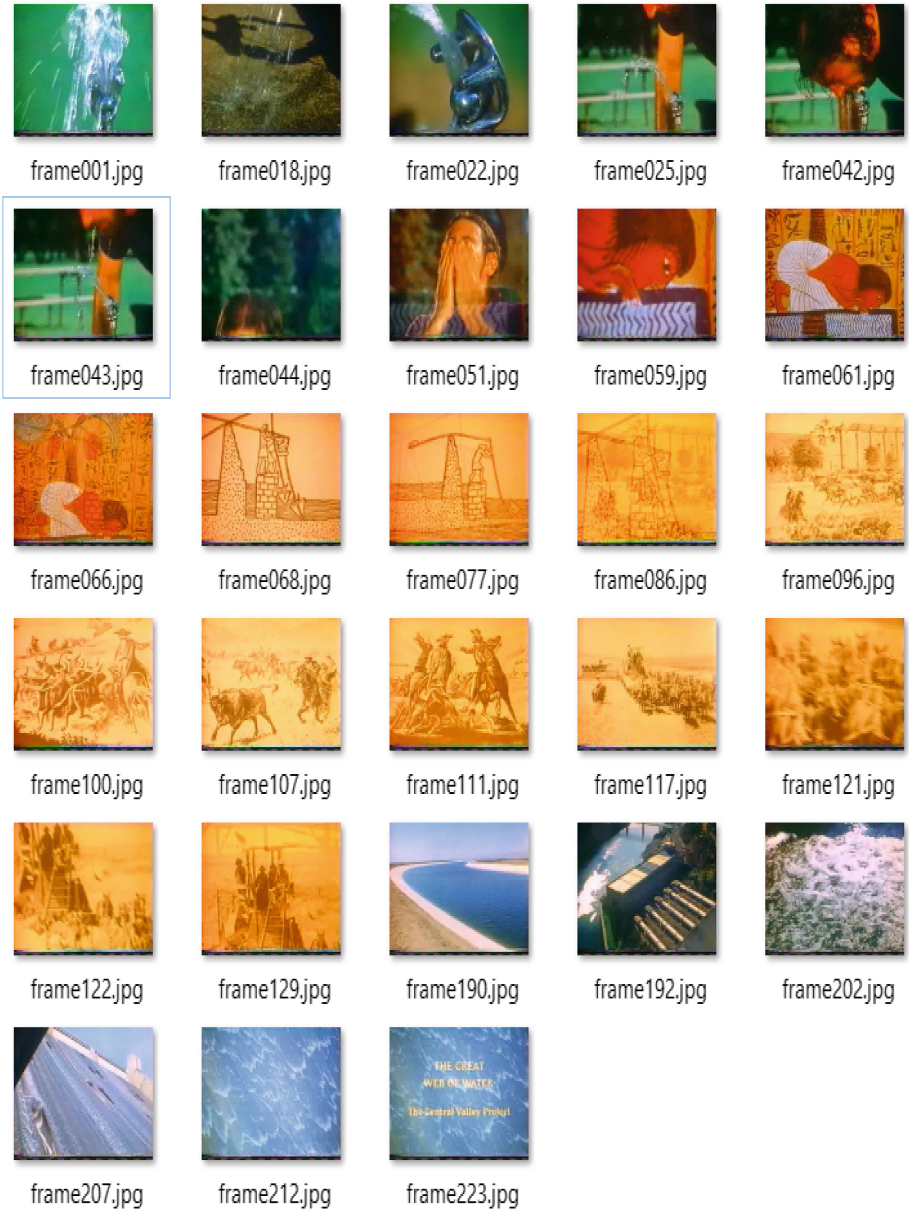

Fig. 8. Video summary result of a sample video from VSUMM dataset

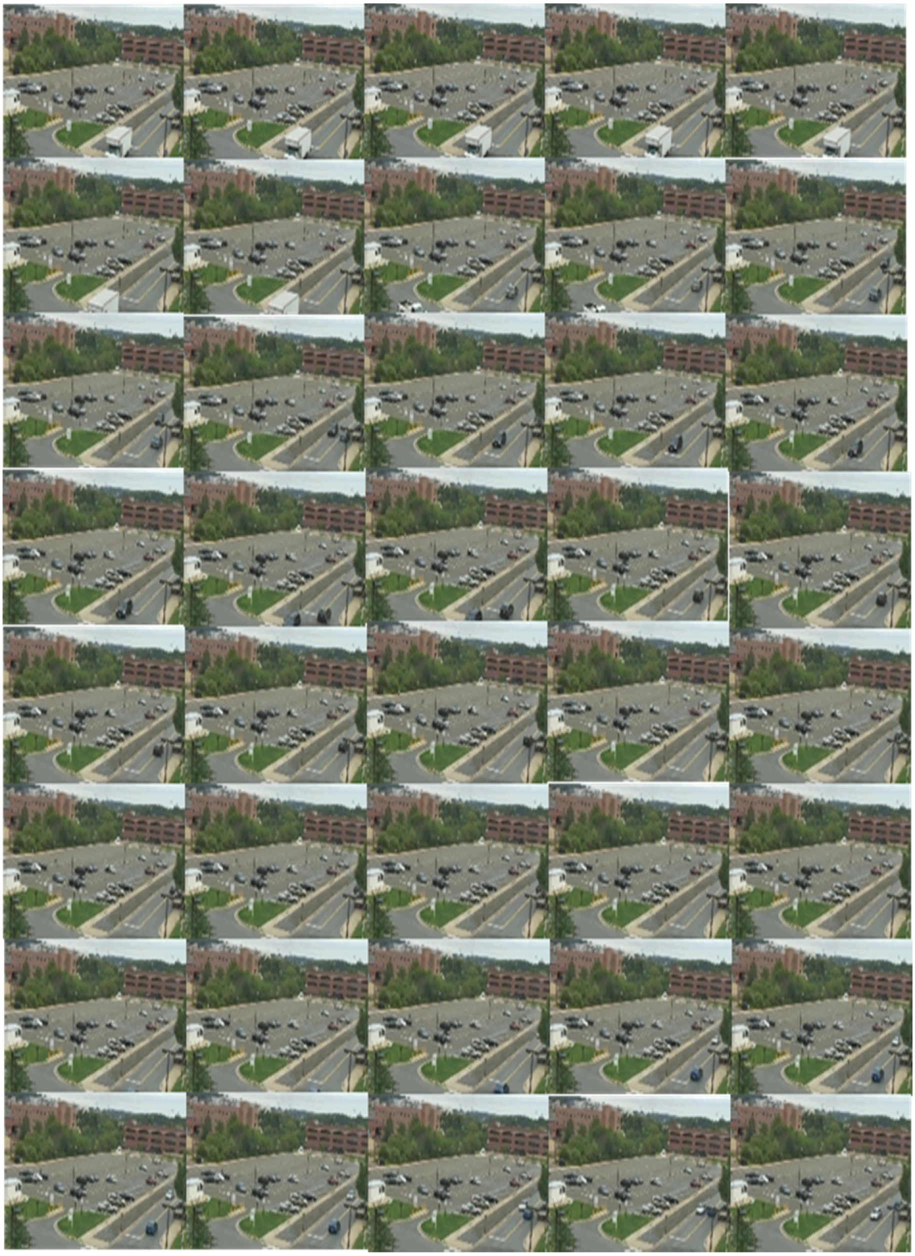

Fig. 9. Video summary result of a sample surveillance video from VIRAT dataset

Table 1. Result analysis using mean opinion score (MOS)

Sr. no.	Sample video	No. of classes	% of frames selected	MOS
1	video-1	5	7.8	4.1
2	video-2	5	6.9	3.8
3	video-3	5	5.5	3.5
4	video-4	5	6.4	4.2
5	video-5	3	4.6	3.6
6	video-6	3	3.2	3.7
7	video-7	3	3.8	3.9

6 Conclusion

This paper proposes a video summarization system called KSUMM that uses visual feature extraction from frames obtained from partial decoding of videos and unsupervised machine learning techniques to perform summarization tasks. The user can tweak the number of classes to be used in the clustering module to have the desired number of frames in the summary. With a 3-class classification, minimum of 3.8% of the frames are included in the summary and 6.4% with a 5-class classification. Number of frames included in the summary increase with the number of classes used in clustering the frames. In this research work, the surveillance video clips compressed using MPEG-4 technology have been used for summary generation. The algorithm for video summarization is designed in such a way that it can be parallelly computed by using graphic processors to speed up the overall process. The quality of the summary is evaluated using Mean Opinion Scores method. The outputs show that the video summary is generated with satisfying Quality of Experience. In future, the technique can be extended to work for other codecs like H.264 and HEVC and deep learning techniques can be employed to replace the manual features with machine learned features.

References

1. Truong, B.T., Venkatesh, S.: Video abstraction: a systematic review and classification. ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) **3**(1), 3 (2007)
2. Babu, R.V., Tom, M., Wadekar, P.: A survey on compressed domain video analysis techniques. Multimedia Tools Appl. **75**(2), 1043–1078 (2014)
3. Kiani, V., Pourreza, H.R.: Flexible soccer video summarization in compressed domain. In: Proceedings of 3rd IEEE International Conference on Computer and Knowledge Engineering, pp. 213–218 (2013)
4. Almeida, J., Leite, N.J., Torres, R.D.S.: Online video summarization on compressed domain. J. Vis. Commun. Image Represent. **24**(6), 729–738 (2013)
5. Almeida, J., Torres, R.D.S., Leite, N.J.: Rapid video summarization on compressed video. In: Proceedings of IEEE International Symposium on Multimedia, pp. 113–120 (2010)

6. Yu, J.C.S., Kankanhalli, M.S., Mulhen, P.: Semantic video summarization in compressed domain MPEG video. In: IEEE International Conference on Multimedia and Expo, Baltimore, MA, USA, 6–9 July, pp. 329–332 (2003)
7. Almeida, J., Leite, N.J., Torres, R.D.S.: VISON: VIdeo Summarization for ONline applications. Pattern Recogn. Lett. **33**(4), 397–409 (2012)
8. Schöffmann, K., Böszörmenyi, L.: Fast segmentation of H.264/AVC bitstreams for on-demand video summarization. In: Satoh, S., Nack, F., Etoh, M. (eds.) MMM 2008. LNCS, vol. 4903, pp. 265–276. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-77409-9_25
9. Herranz, L., Martínez, J.M.: An integrated approach to summarization and adaptation using H.264/MPEG-4 SVC. Sig. Process. Image Commun. **24**(6), 499–509 (2009)
10. Zhong, R., Hu, R., Wang, Z., Wang, S.: Fast synopsis for moving objects using compressed video. IEEE Sig. Process. Lett. **21**(7), 834–838 (2014)
11. Pedregosa, et al.: Scikit-learn: machine learning in Python. JMLR **12**, 2825–2830 (2011)
12. De Avila, S.E.F., Lopes, A.P.B., da Luz Jr., A., de Albuquerque Araújo, A.: VSUMM a mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recogn. Lett. **32**(1), 56–68 (2011)
13. Babu, R.V., Tom, M., Wadekar, P.: A survey on compressed domain video analysis techniques. Multimedia Tools Appl. **75**(2), 1043–1078 (2016)
14. Oh, S., et al.: A large-scale benchmark dataset for event recognition in surveillance video. In: IEEE conference on Computer Vision and Pattern Recognition (CVPR), pp. 3153–3160 (2011)