IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 27, 2018, accepted September 22, 2018, date of publication September 28, 2018,
date of current version November 8, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2872685

Dynamic Summarization of Videos Based on Descriptors in Space-Time Video Volumes and Sparse Autoencoder

JESNA MOHAN[1,2], **(Student member, IEEE), AND MADHU S. NAIR**[3], **(Senior Member, IEEE)**
[1]Department of Computer Science, University of Kerala, Thiruvananthapuram 695581, India
[2]Department of Computer Science, Mar Baselios College of Engineering and Technology, Thiruvananthapuram 695015, India
[3]Department of Computer Science, Cochin University of Science and Technology, Kochi 682022, India

Corresponding author: Jesna Mohan (jesnamohan@gmail.com)

ABSTRACT This paper addresses the problem of generating meaningful summaries from unedited user videos. A framework based on spatiotemporal and high-level features is proposed in this paper to detect the key-shots after segmenting the videos into shots based on motion magnitude. To encode the time-varying characteristics of a video, we explore the local phase quantization feature descriptor from three orthogonal planes (LPQ-TOP). The sparse autoencoder (SAE), an instance of deep learning strategy, is used for the extraction of high-level features from LPQ-TOP descriptors to represent the shots carrying key-contents of videos efficiently. The Chebyshev distance between the feature vectors of the consecutive shots are calculated and thresholded using the mean value of the distance score as the threshold value. The optimal subset of shots with distance score greater than the threshold value is used to generate a high-quality video summary. The method is evaluated using SumMe data set. The summaries thus generated are of better quality than those produced by the other state of the art techniques. The effectiveness of the method is further evaluated by comparing with the human-created summaries in the ground truth.

INDEX TERMS Video Summarization, shot segmentation, LPQ-TOP, sparse autoencoders, Chebyshev distance.

I. INTRODUCTION

Videodata over the internet is increasing day by day due to the advancement of technology. This exponential growth resulted in substantial video repositories where users look through each video to choose an interesting video from it. To alleviate the problem of storage and retrieval of video data, it has become vital to have in place efficient video summarization systems. These systems aim to create abstracts of long videos by detecting and recognizing segments of the video with prime contents and discarding the redundant information. This enables users to select a particular video from a collection of videos by viewing only highlights of the video rather than watching the entire video. Many approaches are proposed for video summarization which can be broadly classified as static summarization methods [1] and dynamic summarization methods [2], [3].

The static summarization methods generate summaries as a set of key-frames by performing frame level analysis. The temporal component of the input video is not preserved in the static summarization. Some of the existing works on static summarization use global features to filter out the key-frames of the video. The frames are then displayed as a sequence in the temporal order to give an overview of entire video to the user. The commonly used global features in literature are color, texture, mutual information, motion information and fuzzy colour histogram [1], [4], [5], [6]. These global features fail to detect localized characteristics of the frames. So, works have been extended using local features [7] of the consecutive frames to identify key-frames. However, a sequence of still images that are isolated and uncorrelated, without any temporal continuation, is not ideal to help the viewer understand the original video.

The dynamic summarization methods generate a subset of original video which preserves temporal component by performing the shot level analysis, where a shot is formed by a set of consecutive frames with the same content. Conventional approaches for dynamic summarization concentrate on extracting prime features that are capable of discarding redundant frames, thereby preserving the essential contents of the video. Hu *et al.* [8] emphasized the use of importance score estimated from features based on visual attention. Then, the video clips with higher importance score are included in the summary. Hu and Li [9] combined global and local features to select an optimal subset of meaningful shots from the video. Zhao and Xing [10] proposed a dictionary-based method using sparse coding, which generates the summary by combining the segments which cannot be reconstructed using a learned dictionary. A generic video summarization method is presented in [2], wherein the video is first segmented into shots based on their motion magnitude. Then, the shots are assigned an interesting score which is computed based on specific features. The frames with the highest importance score are chosen to create summaries.

Now, these methods have been extended by capturing visually significant portions of videos [11]. The user attention models are build based on saliency representations, using multimodal features by combining audio, visual and textual information. Then, the saliency curves are used to find the points where peak attention is attained. Recently, Fei *et al.* [12] proposed a method giving significance to memorability score, predicted using Hybrid-AlexNet. The score is combined with motion cues to determine key shots. The selection of key shots is also modelled using optimization techniques such as Particle Swarm Optimisation(PSO) as in [13] and [14]. A multiobjective energy function including interestingness and representativeness of frames is used in [15] to rank the frames based on submodular maximisation technique. The summarization of the user videos is done by capturing the object-level features from the videos. Lee and Grauman [16] incorporated web images as prior information to select informative portions from the user videos without using any human annotated summaries. These video summarization methods have also been extended to semantic level processing based on deep networks [17], [18], [19]. The deep features have more power than handcrafted features, to discriminate between the relevant and irrelevant content.

Most of the works in literature has focused on domain dependent approaches where the method is fine-tuned to capture domain specific characteristics [20], [21]. The existing approaches concentrate mainly on edited videos such as sports, news, cartoon etc. that has a specific structure. With the prominence of unedited user videos due to the development of electronic gadgets, it became necessary to design methods to handle such videos, where previous research methods cannot be applied directly.

Moreover, Lee *et al.* [22] proposed a method to predict the importance for each frame using linear regression model based on the saliency of frames. However, this method is only applicable to videos captured using the wearable camera. Some existing approaches in [23], [24], and [25] generated summaries using a panoramic image formed from a few consecutive significant frames in the input video. Recently, Chen *et al.* [26] proposed a method based on spectrum analysis for generating the summaries from traffic videos.

Inspired by the demand for developing more accurate and robust domain-independent methods for summarizing user videos, we present a generic framework for creating dynamic summaries from these videos. The user videos are raw, unedited and long videos, often containing an interesting event. These videos run for several hours and are summarized using object-centred approaches. But user videos include many events in a single video and features based on a particular object in the video is insufficient. So, we introduce a method that focuses on extracting high-level features which aid in understanding the semantic content of videos. It first segments the input video into shots using shot segmentation algorithm based on motion magnitude between consecutive frames. Each shot of the video is processed as space-time video volume and then the feature vectors are extracted from each volume using Local Phase Quantization from Three Orthogonal Planes(LPQ-TOP). A high-level representation of these feature vectors is generated using the encoding part of a Sparse Autoencoder (SAE). The Chebyshev distance vector between consecutive frames is thresholded to generate the final summary. The generated video summaries can represent the key contents in less time removing the redundant information from the input video. Experimental results of the proposed method on SumMe dataset, which is the benchmark dataset for dynamic video summarization, demonstrate the effectiveness of this method.

The rest of this paper is organized as follows. Section II describes the proposed methodology for generating dynamic summaries from input video. Experimental results and discussions are illustrated in Section III. We conclude the paper in Section IV.

II. PROPOSED METHODOLOGY

The proposed method is firmly built on a framework based on spatiotemporal and low-level features which can efficiently generate dynamic summaries from input videos. The generated summaries are based on shots corresponding to the input video driven by distances between feature vectors of consecutive shots. FIGURE 1 gives an overview of the proposed approach. The detailed steps are as follows.

A. SHOT SEGMENTATION

Shot segmentation is an essential step in summarization since the quality of the generated summary depends on segmented shots from the input video. The shot segmentation methods divide the frames of the input video into a subset of frames where each subset consist of a set of consecutive frames that are similar. The first and the last frames of each subset represent a content change between the shots. As far as the video is considered, a significant content change can be captured by monitoring the magnitude of motion vectors between the consecutive frames. The proposed method utilizes the superframe segmentation technique in [2]. The proposed method is a robust technique to segment unedited user

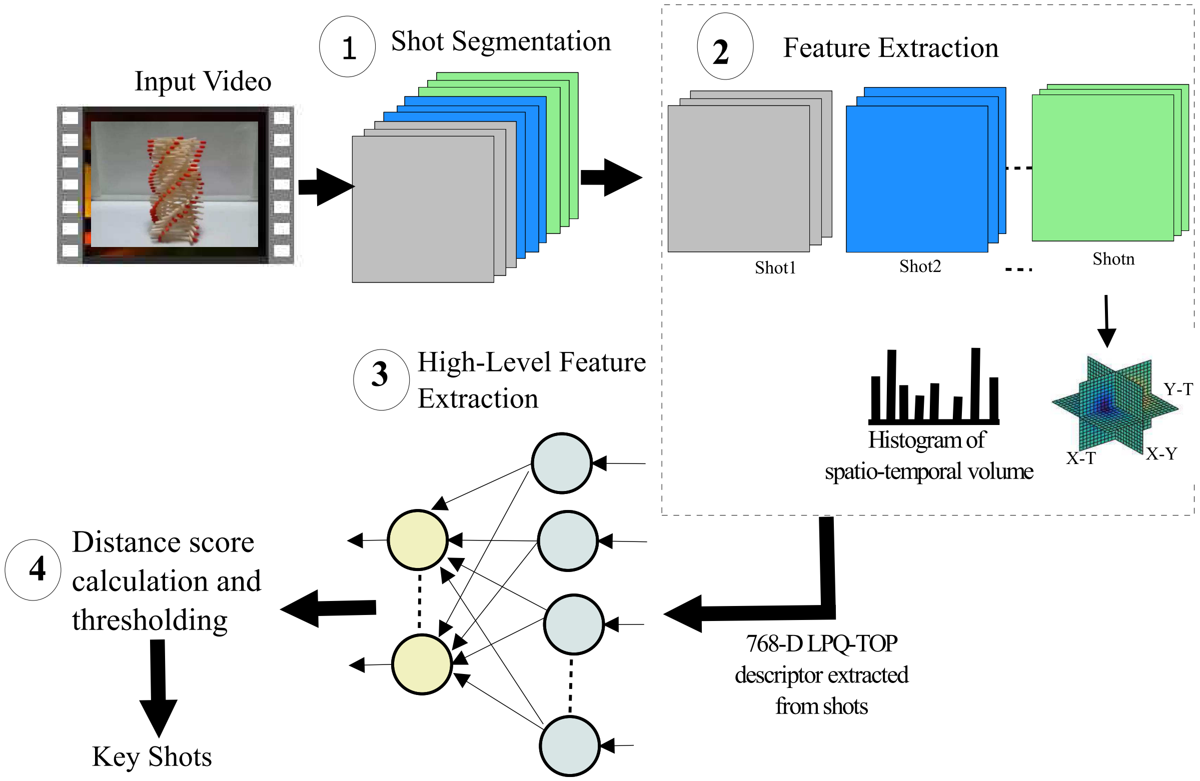

1 Shot Segmentation
Input Video
2 Feature Extraction
Shot1
Shot2
Shotn
3 High-Level Feature Extraction
Histogram of spatio-temporal volume
Y-T
X-T
X-Y
4 Distance score calculation and thresholding
768-D LPQ-TOP descriptor extracted from shots
Key Shots

FIGURE 1. Overview of the proposed method.

videos, where the commonly used segmentation algorithms based on histograms, color histograms etc. fail. The method detects the scene change between frames by optimizing an energy function based on the magnitude of motion between consecutive frames in a video. The energy function is given by (1).

$$E(S_j) = \frac{1}{1+\gamma C_{cut}(S_j)}.P_l(\mid S_j \mid) \quad (1)$$

where cut cost is denoted by C_{cut}, P_l is the length prior for superframes, $|.|$ is the length of superframe and γ controls the influence between the cut cost and the length prior. The cut cost is calculated using (2).

$$C_{cut}(S_j) = m_{in}(S_j) + m_{out}(S_j) \quad (2)$$

The motion magnitude of the first and the last frame of the superframe is denoted as $m_{in}(S_j)$ and $m_{out}(S_j)$. It is computed as the average magnitude of the forward and the backward motion based flow vectors which is estimated using Kanade–Lucas–Tomasi (KLT) feature tracker. The cut cost is lower for frames that correspond to no motion or with less motion and higher for frames with a significant change in motion. The histogram of the segment lengths of user-created summaries in the dataset is computed and the lognormal distribution is fitted to histogram to find its peak value. The maximum segment length is chosen to be the best length which is represented by length prior P_l. The shots corresponding to the input video are initialized by dividing the set of frames based on the length prior. The boundaries are iteratively updated by optimizing the energy function in (1) using the hill-climbing optimization algorithm in [27].

B. SHOT-LEVEL FEATURE EXTRACTION

The next step in the proposed method is the extraction of feature descriptors from the shots. Since the video consists of both spatial and temporal components, the spatiotemporal features of each shot extracted in the previous step are used for further processing. We explore LPQ-TOP which is the temporal extension of Local Phase Quantization (LPQ) that represent dynamic image texture efficiently. The descriptor has already been applied successfully in emotion recognition from videos [28]. The resultant vector combines both temporal and appearance characteristics of the frames in the shots.

1) LOCAL PHASE QUANTIZATION DESCRIPTOR FROM THREE ORTHOGONAL PLANES (LPQ-TOP)

Ojansivu and Heikkila proposed a very robust texture feature, Local Phase Quantization (LPQ) operator in [29]. The descriptor is extracted from the short-term Fourier transform (STFT) representation of the input image instead of computing descriptors from raw pixel values. The descriptor is calculated at each pixel position in the image by defining a small neighbourhood and considering its phase information for computing the feature values. The Fourier transform defined over the neighbourhood N_x at the pixel position x is computed using (3).

$$F(u,x)=\sum_{y\varepsilon N_x} f(x-y)e^{-j2\pi u^T y}={w_u}^T f_x \tag{3}$$

where basis vector of 2-D FDT at frequency u is represented by w_u and vector f_x contain all samples from N_x. The feature vector corresponding to a pixel position has four components each representing Fourier coefficients at four points in the frequency domain. The four points considered are $a_1 = [1,0]^T, a_1 = [0,1]^T, a_1 = [1,1]^T, a_1 = [1,-1]^T$. The signs of real and imaginary parts of Fourier coefficients are quantized using a scalar quantiser to generate eight-bit binary coefficients. The quantizer will replace negative values to '0' and positive values to '1' and binary coding using (4) is done to find integers. The 256-D LPQ feature vector is then computed from the histogram of integer values corresponding to the pixel location.

$$F(u,x)=\sum_{i=1}^{8} q_i 2^{i-1} \tag{4}$$

The LPQ features from Three Orthogonal Planes, XY, XT and YT, are concatenated to form 768-D($256 \times 3 = 768$) LPQ-TOP descriptors per space-time volume. Suppose a shot is of size $M \times N \times 45$. So there are 45 frames in one shot and the size of each frame in the shot is $M \times N$. Let, the size of the neighbourhood be $w_x \times w_y$ in the XY plane, $w_y \times w_t$ in the YT plane and $w_x \times w_t$ in the XT plane. Therefore LPQ-TOP is calculated at pixel position $p = (x, y, t)$ based on the central pixel (x_c, y_c, t_c) as follows

The LPQ-TOP in XY plane is calculated by considering pixel positions such that

$$x\varepsilon\{x_c-\frac{(w_x-1)}{2} \text{ to } x_c+\frac{(w_x-1)}{2}\}$$
$$y\varepsilon\{y_c-\frac{(w_y-1)}{2} \text{ to } y_c+\frac{(w_y-1)}{2}\}$$
$$t=t_c$$

Similarly, for XT plane, the pixel positions are

$$x\varepsilon\{x_c-\frac{(w_x-1)}{2} \text{ to } x_c+\frac{(w_x-1)}{2}\}$$
$$y=y_c$$
$$t\varepsilon\{t_c-\frac{(w_t-1)}{2} \text{ to } t_c+\frac{(w_t-1)}{2}\}$$

and for YT plane, the pixel positions are

$$x=x_c$$
$$y\varepsilon\{y_c-\frac{(w_y-1)}{2} \text{ to } y_c+\frac{(w_y-1)}{2}\}$$
$$t\varepsilon\{t_c-\frac{(w_t-1)}{2} \text{ to } t_c+\frac{(w_t-1)}{2}\}$$

The histogram is calculated from spatiotemporal volume as in (5).

$$H_{i,j}=\sum_{x,y,t} I(f(x,y,t)=i), \quad i=0,1,\ldots..,255,\ j=0,1,2 \tag{5}$$

where, the code value of LPQ corresponding to pixel (x, y, t) in the j^{th} plane is $f(x, y, t)$. The histogram is normalized and concatenated to form the final descriptor.

C. HIGH-LEVEL FEATURE EXTRACTION

High-level representation of the spatiotemporal feature vectors are generated using SAE in this step. We input hand-crafted LPQ-TOP descriptors of the shots to SAE, which encode them to high-level ones. Autoencoders, an instance of the deep learning strategy, works by minimizing the reconstruction error using the backpropagation algorithm. The network learns a set of weights corresponding to the data, after the convergence. SAE represents the high dimensional input vectors using low dimensional vectors and are used as an alternative to the dimensionality reduction technique such as Principal Component Analysis (PCA).

The three main layers of a basic autoencoder are input layer, encoding layers and decoding layer. Suppose x is an input vector to the autoencoder such that it is an element of d - dimensional space ($x \in R^d$). The output z corresponding to x belongs to k - dimensional space ($z \in R^k$) such that $k < d$ and is given in (6).

$$z=\sigma(Wx+b) \tag{6}$$

where the weight matrix is $W \in R^{d\times k}$ and the bias for encoding is $b \in R^d$. The function σ can be a ReLu (Rectified Linear Unit) function or a sigmoid, which is a differentiable function. The reconstruction error between the input and output is given by (7).

$$error=\sum_{r=1}^{n}\frac{1}{2}\left\|\hat{x}^{(r)}-x^{(r)}\right\|^2 \tag{7}$$

where n is the number of training samples. SAE is a new variant of the autoencoder in which additional constraints are imposed on the network to avoid the overfitting problem. To prevent the overfitting, sparsity regularization constraint is added to the hidden layer. The loss function of the hidden layer 'h' of SAE layer is given by (8).

$$J_{sparse}(w_h,b_h,\hat{w}^h,\hat{b}^h)+\beta\sum_{j=1}^{n_h} KL(\rho \parallel \hat{\rho}_j) \tag{8}$$

where $J_{sparse}(w_h, b_h, \hat{w}^h, \hat{b}^h)$ is calculated as in (9).

$$J_{sparse}(w_h,b_h,\hat{w}^h,\hat{b}^h)=\frac{1}{2}\sum_{i=1}^{n}\left\|\hat{h}^{(i)}-x^{(i)}\right\|_2^2+\frac{\lambda}{2}\|w_h\|^2 \tag{9}$$

The weight decay parameter is denoted by 'λ', sparsity penalty weight is denoted by β and 'n_h' denotes the number of neurons in the hidden layer 'h'. The input to the hidden layer 'h' is same as the output of the hidden layer $(h-1)$. The second term in (8) is the Kullback-Leibler Divergence (KL Divergence) which is the penalty term added to make the activations of latent units close to zero. The KL Divergence is given by (10).

$$KL(\rho \parallel \hat{\rho}_j) = \rho \, log\frac{\rho}{\hat{\rho}_j} + (1-\rho)log\frac{(1-\rho)}{\hat{\rho}_j} \qquad (10)$$

where the sparsity parameter is denoted as ρ and $\hat{\rho}_j$ represents the average activation of the hidden unit. The architecture of SAE used in the proposed approach is shown in FIGURE 2, which consists of 2 hidden layers. The proposed architecture used only the encoding layer of SAE.

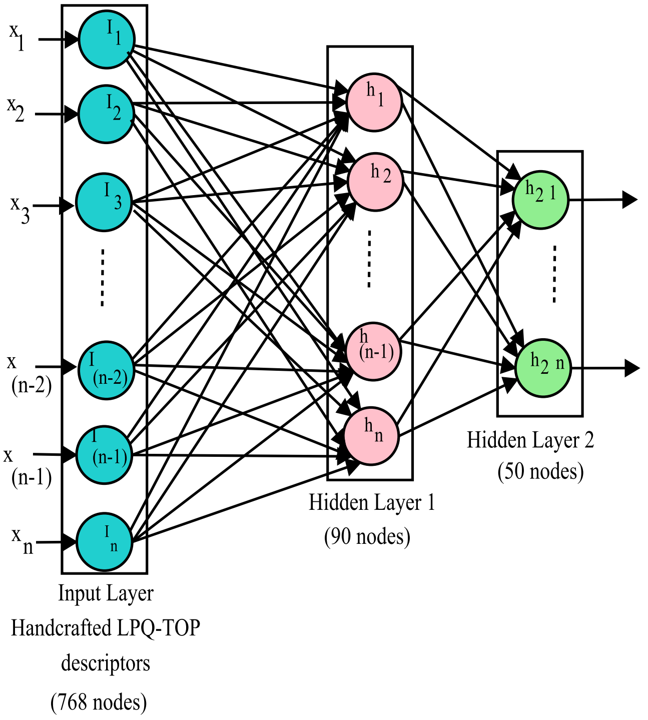


FIGURE 2. Architecture of SAE used in the proposed method.

D. DETECTING KEY SHOTS

The next step in the proposed approach is to find the similarity between the feature vectors corresponding to consecutive shots. The important step in video summarization is the choice of the distance measure and the threshold value to determine key-shots. An appropriate distance measure plays a significant role in content-based image retrieval system [30]. Here, we explored Chebyshev distance score between feature vectors of the consecutive shots as a measure of similarity between the shots. The distance measure is then thresholded to filter out key shots. The distance between high-level feature vectors extracted using SAE from LPQ-TOP descriptors of consecutive shots are then computed. To discriminate content change between the shots, the Chebyshev distance between deep features of the onsecutive frames is chosen as the best distance metric. The distance metric is chosen based on the analysis done on the impact of the metric on the summary generated by the algorithm. The more similar shots have low distance score between them. If FV_1 and FV_2 are n-dimensional feature vectors corresponding to two shots with FV_1=$\{u_1, u_2, ...u_n\}$ and FV_2=$\{v_1, v_2, ...v_n\}$, where i=1, 2, ..., n, then, Chebyshev distance is calculated using (11).

$$Chebyshev\ distance(FV_1, FV_2) = \max_i(|u_i - v_i|) \qquad (11)$$

Let D_F represents the set of Chebyshev distance scores between feature vectors of the consecutive shots. The displacement magnitude between shots in D_F is denoted as $d_1, d_2, d_3 \ ..d_{n_s-1}$. The shots whose distance score greater than a threshold value are selected as key shots since there is a dissimilarity in the contents represented by the frames as reflected by the distance values of feature vectors between the selected shot and the next shot. The mean value of entire distance array is taken as the threshold value which is determined empirically. Suppose T_{thresh} be the mean value obtained from the set of displacement values in D_F. All shots with displacement value greater than T_{thresh} is added to the final set of key shots V_K. FIGURE 3 illustrates the magnitude of distance score between consecutive shots of 'Fire Domino.webm' video in the SumMe dataset and the straight line shows the mean value of distance score, which is chosen as threshold value to detect the shots to be included in the summary.

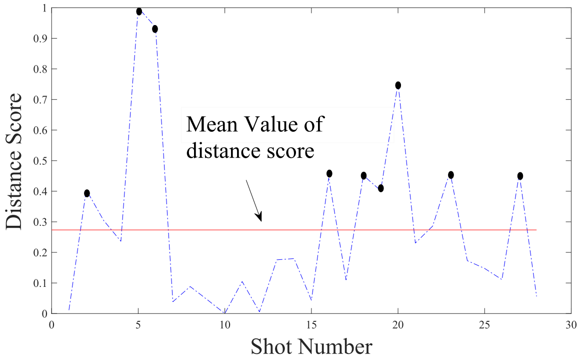


FIGURE 3. Magnitude of distance score between consecutive shots.

III. EXPERIMENTAL ANALYSIS

Experiments have been performed on publicly available SumMe dataset which contains 25 user videos of different categories belonging to egocentric, moving and static videos. The duration of each video ranges from 1 to 6 minutes. After human evaluations, the frames of these videos in the dataset are marked as interesting or not. Atleast 15 human summaries corresponding to each video is given in the dataset. As the ground truth consists of summaries whose length is set to be 15% of the length of the video, we have chosen the same summary length in our experiments.

All implementation is done in MATLAB on Windows 10 Pro with an Intel(R) Core(TM) i7-3770 CPU at 3.40GHz with 4.00GB RAM running 64-bit operating system.

A. PERFORMANCE METRICS

There is no consistent evaluation metrics in video sumarization since there is no objective ground truth in summarization.

Two abstracts of the same video cannot be compared even by humans because some parts of the video which seek attention of one user may not be attractive to the other. The effectiveness and efficiency of the proposed approach is evaluated using Precision, Recall and F-score. The evaluation metrics are computed based on similarity between the frames in the output with those in the user summaries of the dataset. The evaluation metrics are calculated as follows.

Let N_{total} represents the total number of frames in the input video,

N represents the total number of frames in the output,

N_{nm} represents the number of non-matching frames in the output compared to the frames in the ground truth.

N_m represents the number of matching frames in the output compared to the frames in the ground truth and

N_{GT} represents the number of frames in the ground truth. We can then define,

$$Precision = \frac{N_m}{N} \tag{12}$$

$$Recall = \frac{N_m}{N_{GT}} \tag{13}$$

$$\text{F-score} = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \tag{14}$$

The evaluation metrics are calculated separately for each user summary in the dataset. The final F-score is calculated by finding the average of these scores. If there are n users in the dataset, F-score is calculated as

$$Overall\ \text{F-score} = \frac{\sum_{i=1}^{n} \text{F-score}_i}{n} \tag{15}$$

B. RESULTS AND DISCUSSIONS

We evaluated the quality of generated summaries using proposed method by comparing with the human created summaries in the ground truth. In particular, the algorithm first converts each video into shots using shot segmentation algorithm based on motion magnitude with the parameter γ set to 1, δ with the initial value set to 0.25 seconds. Motivated by two recent summarization works [31], [12], we used a combination of handcrafted spatiotemporal features and high-level features to improve the accuracy of summarization step. After performing segmentation, LPQ-TOP features are extracted from the shots. The window size used for Fourier phase computation is set to [5,5] for LPQ-TOP features and we chose the same window size for the computation of LPQ descriptors in XY, XT and YT planes. The histogram of LPQ descriptors with 256 bins from the three orthogonal planes are combined to form 768-D LPQ-TOP descriptor which is passed to SAE for generating a high-level representation. The subsequent sections deal with finding the number of hidden nodes in SAE, the parameter values used in SAE, results obtained through the proposed method and also the comparison of the proposed method with the other state-of-the-art methods.

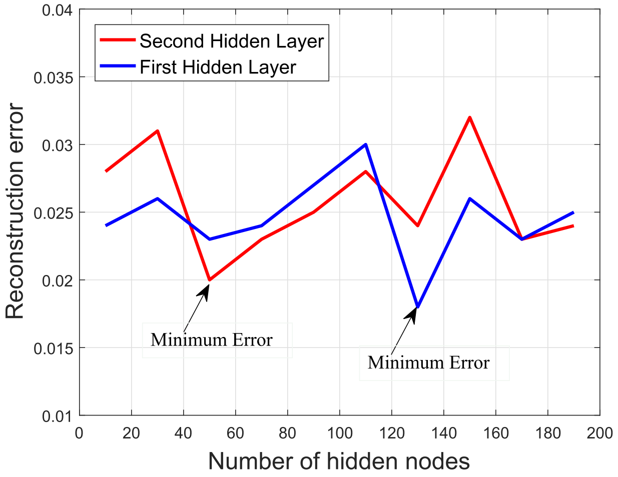


FIGURE 4. Comparison of the number of hidden nodes in each layer and reconstruction error.

1) FINDING NUMBER OF HIDDEN NODES OF SAE

There are no optimal techniques for choosing the number of hidden layers [32]. Trial and error methods are usually used for selecting the number of neurons [33]. The decision of the number of nodes in the hidden layer is crucial as it influences the components of the feature vector. Here, we extracted the reduced feature vector from the last hidden layer of SAE with two hidden layers. The dimensionality of the feature vector is equal to the number of nodes in the last hidden layer. The number of nodes is chosen so that the reconstruction error between the input and the output calculated as in (7), is minimum. FIGURE 4 shows the plot of the number of hidden nodes versus reconstruction error on SumMe dataset. Based on the plot, the number of nodes in the first hidden layer is chosen to be 130 and that of second hidden layer to be 50 nodes. The number of nodes in the input layer is 768 which is equal to the dimension of feature vectors. We have chosen the number of nodes in hidden layers from [10,30,50,70,90,110,130,150,170,190]. The number of nodes in the second hidden layer is plotted in the curve with fixed value for the number of nodes in the first hidden layer (130 nodes which corresponds to minimum reconstruction error).

2) PARAMETER SETTING OF SAE

This section gives the values of six parameters that affect the performance of SAE. The values of the parameters are chosen as in [34] such that Sparsity penalty (β)=3, Sparsity proportion (ρ)=0.05, Weight decay penalty (λ)=0.003, Convergence tolerance (γ)=1e-9 and the Maximum number of iterations (δ)=400. The choice of activation function is also very important in SAE. Based on preliminary experiments conducted here, we selected the sigmoid function as it gave better accuracy than ReLu function.

3) RESULTS OF THE PROPOSED METHOD

TABLE 1 illustrates the results of the proposed method on each video in SumMe dataset. It gives the category of each

TABLE 1. Results of the proposed method for each video in SumMe dataset. *E* - egocentric videos, *M* - moving videos, *S* - static videos.

Category	Video name	NF_{in}	NF_{out}	NS_{in}	NS_{output}	RR	Precision	Recall	F-score
E	Base jumping	4729	2454	84	63	0.481	0.497	0.186	0.277
E	Bike Polo	3064	1625	55	49	0.469	0.760	0.321	0.421
E	Scuba	2221	954	40	22	0.570	0.408	0.130	0.210
E	Valparasio downhill	5178	2719	92	73	0.475	0.535	0.246	0.346
M	Bearpark climbing	3341	1997	60	51	0.402	0.120	0.029	0.055
M	Bus in rock tunnel	5131	1984	92	44	0.613	0.218	0.044	0.082
M	Car railcrossing	5075	1486	91	42	0.707	0.417	0.109	0.182
M	Cockpit landing	9046	4644	162	122	0.487	0.092	0.032	0.061
M	Cooking	1286	348	23	11	0.729	0.451	0.200	0.311
M	Eiffel tower	4971	1785	89	62	0.641	0.743	0.240	0.342
M	Excavators river crossing	9721	4829	174	132	0.503	0.397	0.184	0.280
M	Jumps	950	259	17	5	0.727	0.695	0.398	0.540
M	Kids playing in leaves	3187	1648	57	42	0.483	0.459	0.178	0.276
M	Playing on water slide	3065	1729	55	46	0.436	0.053	0.008	0.016
M	Saving dolphines	6683	4651	119	103	0.304	0.251	0.079	0.138
M	St. Maartenlanding	1751	614	31	13	0.649	0.810	0.466	0.590
M	Statue of Liberty	3863	2239	69	57	0.420	0.344	0.084	0.144
M	Uncut evening flight	9672	4817	173	151	0.502	0.568	0.318	0.436
M	Paluma jump	2574	829	46	24	0.678	0.558	0.273	0.394
M	Playing ball	3120	1540	56	41	0.506	0.415	0.163	0.250
M	Notre Dame	4608	2841	82	70	0.383	0.378	0.147	0.234
S	Air Force one	4494	2666	80	68	0.407	0.706	0.366	0.461
S	Fire Domino	1612	877	29	23	0.456	0.682	0.392	0.511
S	Car over camera	4382	840	78	19	0.808	0.618	0.188	0.295
S	Paint ball	6096	2360	109	69	0.613	0.666	0.333	0.456
S	Mean					0.538(53.8%)	0.474	0.205	0.292

video along with the name of the video, the number of frames in the input video (NF_{in}), the total number of output frames (NF_{out}), the number of shots corresponding to input video (NS_{in}), the number of shots in the output (NS_{output}), Reduction rate (RR), Precision, Recall and F-score corresponding to each video. RR is calculated as the ratio of the number of frames in the output to the number of frames in the input. The results show that the proposed method attain an average reduction rate of about 50% with F-score of 0.292. In the experiments, we evaluate our method using summaries in the ground truth. For this, average F-score of each human summary is calculated by comparing the summary generated by one particular user with those of the other users which measures the performance of human summaries. The average F-score of the human summaries in SumMe dataset is 0.311 as given in TABLE 2. So, the proposed method achieved an accuracy of 94.1% relative to the average human score.

We also conducted experiments to evaluate the performance of LPQ-TOP features and high-level features from SAE. The LPQ-TOP features alone achieved a F-score of 0.1742 and an accuracy of 56.01% relative to the average human score. The comparison of performance is given in FIGURE 5. It is clear from the plot that combining LPQ-TOP features with features from SAE provided considerable increase in the accuracy.

4) COMPARISON WITH THE OTHER STATE-OF-THE-ART METHODS

In order to validate our video summary evaluation method, we conducted a comparative study of its performance. We focused on evaluating six video summarization techniques using benchmark summaries in the SumMe dataset. We used automatic summaries from existing summarization techniques such as Uniform Sampling (US) which selects video segments at uniform intervals to include in the summary, Clustering based method (CLUST) in [22] which finds key-segments by clustering of colour histogram, Visual attention based method (ATTEN) in [35] which explore principles of visual saliency to find important parts of video, interestingness score based method (SUMME) in [2], Spatiotemporal based method in [31] which combines spatial and temporal features to assign a spatiotemporal score to each segment and significant segments are selected based on this score, and semantic features based method (SEMANTIC) in [12]. TABLE 2. shows the results of the comparative analysis of the proposed method with the other summarization techniques. It shows the F-score of each video in the dataset using the different techniques and the F-score values of human-generated summaries which is denoted as GT. The F-score values of human-generated summaries of a video measure the human performance which is computed by comparing one user summary with summaries generated by the other users. The average F-measure of these user-created summaries is calculated and is compared with the F-score of the proposed method and the F-score of other methods as given in TABLE 2. The results show that our method with an average performance of 94.1% performs better than the recent approach based on semantic features with an average performance of 90.3%. FIGURE 6 shows the performance of different categories of videos in SumMe dataset and its comparison with other methods. The results show that our method attains better results for egocentric and static videos with F-score of moving videos slightly lower but close to

TABLE 2. Results of various categories of videos in SumMe dataset. E - egocentric videos, M - moving videos, S - static videos, uniform sampling (US), clustering based method (CLUST), visual attention based method (ATTEN), interestingness score based method (SUMME), spatiotemporal based method (S-T), semantic features based method (SEMANTIC).

Category	Video name	GT	US	CLUST	ATTEN	SUMME	S-T	SEMANTIC	Ours
E	Base jumping	0.257	0.117	0.109	0.163	0.121	0.205	0.292	0.277
E	Bike Polo	0.322	0.195	0.130	0.094	0.352	0.191	0.289	0.421
E	Scuba	0.217	0.079	0.135	0.227	0.184	0.204	0.222	0.210
E	Valparasio downhill	0.272	0.193	0.154	0.212	0.242	0.250	0.215	0.346
M	Bearpark climbing	0.208	0.101	0.158	0.155	0.118	0.268	0.191	0.055
M	Bus in rock tunnel	0.198	0.133	0.102	0.151	0.135	0.212	0.303	0.082
M	Car railcrossing	0.357	0.134	0.146	0.058	0.362	0.144	0.195	0.182
M	Cockpit landing	0.279	0.106	0.156	0.143	0.172	0.176	0.334	0.061
M	Cooking	0.379	0.135	0.139	0.071	0.321	0.369	0.272	0.311
M	Eiffel tower	0.312	0.139	0.179	0.088	0.295	0.226	0.206	0.342
M	Excavators river crossing	0.303	0.170	0.163	0.043	0.189	0.212	0.149	0.280
M	Jumps	0.483	0.013	0.298	0.165	0.427	0.548	0.424	0.540
M	Kids playing in leaves	0.289	0.114	0.165	0.169	0.089	0.184	0.317	0.276
M	Playing on water slide	0.195	0.081	0.141	0.114	0.200	0.164	0.260	0.016
M	Saving dolphines	0.188	0.133	0.214	0.174	0.145	0.219	0.189	0.138
M	St. Maartenlanding	0.496	0.121	0.096	0.397	0.313	0.354	0.522	0.590
M	Statue of Liberty	0.184	0.128	0.125	0.059	0.192	0.159	0.153	0.144
M	Uncut evening flight	0.350	0.136	0.098	0.283	0.271	0.327	0.367	0.436
M	Paluma jump	0.509	0.221	0.072	0.066	0.181	0.193	0.317	0.190
M	Playing ball	0.277	0.128	0.176	0.132	0.174	0.191	0.179	0.250
M	Notre Dame	0.231	0.205	0.141	0.119	0.235	0.259	0.173	0.234
S	Air Force one	0.332	0.057	0.143	0.215	0.318	0.285	0.437	0.461
S	Fire Domino	0.394	0.148	0.349	0.258	0.130	0.354	0.318	0.511
S	Car over camera	0.346	0.129	0.206	0.111	0.372	0.303	0.380	0.295
S	Paint ball	0.399	0.071	0.198	0.292	0.320	0.417	0.346	0.456
	Mean	0.311	0.129	0.163	0.158	0.234	0.257	0.281	0.292
	Relative to average human	100%	41.5%	53%	51%	75%	82.6%	90.4%	94.1%

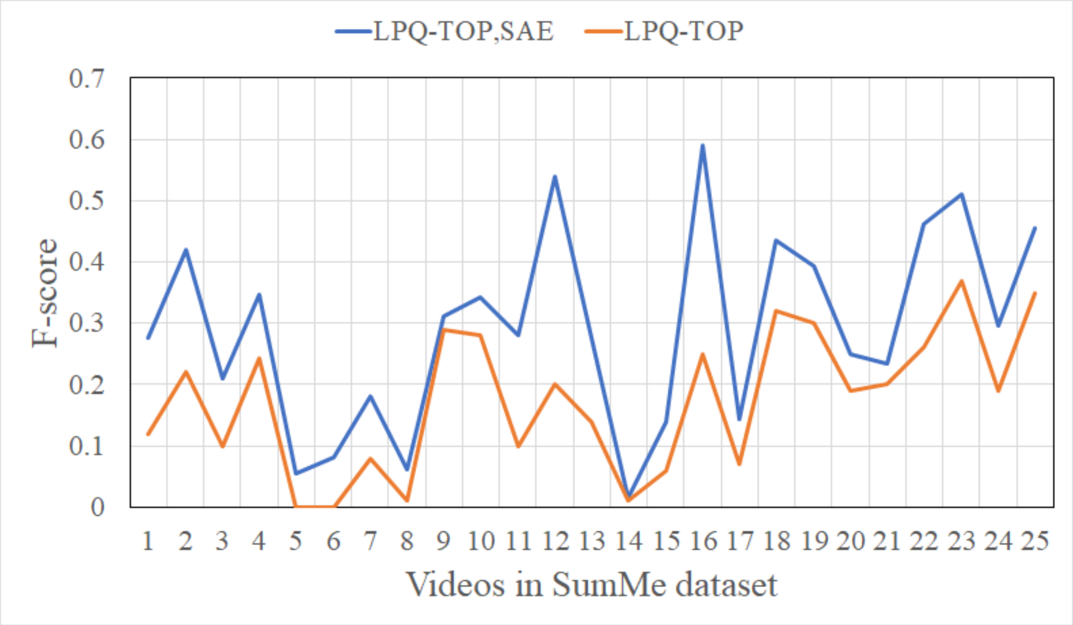


FIGURE 5. Evaluation of handcrafted features and high level features.

the other state-of-the-art methods. This is because in static videos the movement of objects in consecutive frames is lesser compared to other types of videos.

FIGURE 7 illustrates the sample output of summaries generated by the proposed method of three videos, one video from each category as a plot with the frame number on the X-axis and human-generated score on the Y-axis for the selected frames as given in the ground truth of the dataset. For a given frame, the score is calculated as the ratio of the number of users who has selected the particular frame as output to

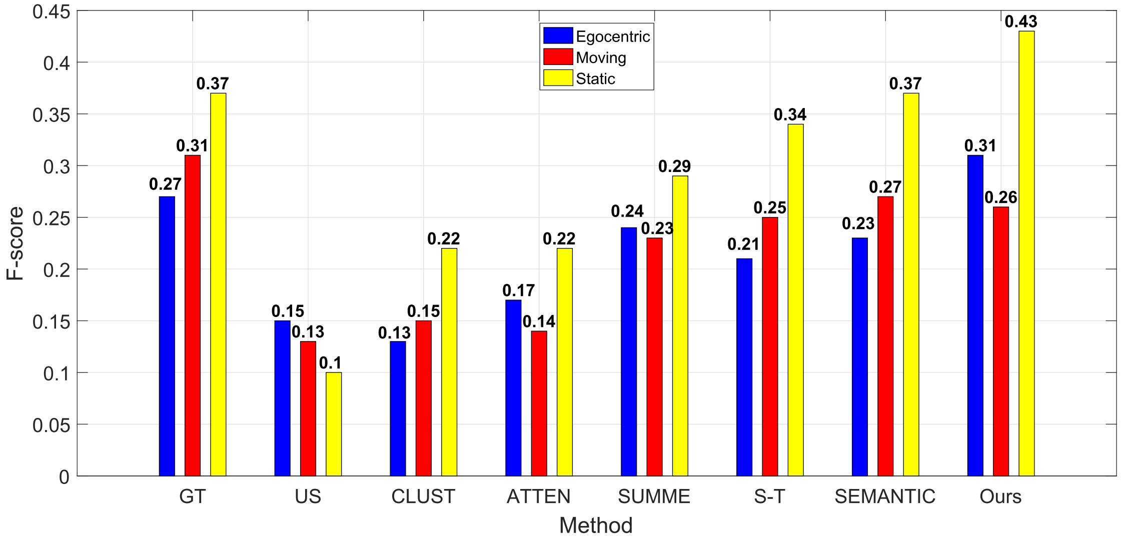
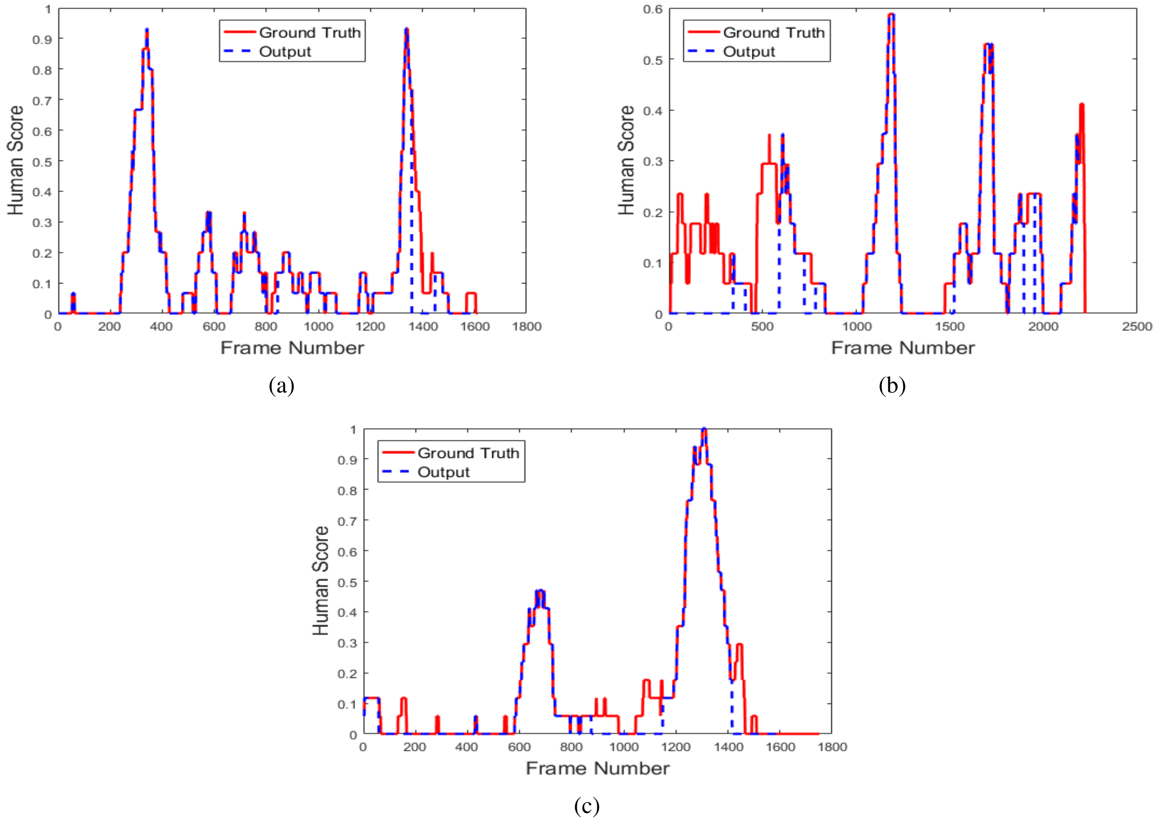


FIGURE 6. Comparison of Summaries generated by the proposed method with the other state-of-the-art summarization methods.



(a)

(b)

(c)

FIGURE 7. Comparison of summaries generated by the proposed method with the human generated summaries. (a) Static video 'Fire Domino'. (b) Egocentric video 'Scuba'. (c) Moving video 'St Maarten Landing'.

the total number of users. If the total number of users is 15 and the number of users who selected the frame is 1, the score is 0.066 ($\frac{1}{15}$), the score is 0 if none of the users selected the frame and 1 if all the users selected the frame as key frame. FIGURE 7 (a - c) shows the plot of human-generated summaries and the output of the proposed method for the static video 'Fire Domino', egocentric video 'Scuba' and moving video 'St Maarten Landing', respectively. The overlapping blue and red line shows that the output generated by the proposed approach is similar to those in the ground truth. The frames shown using solid red line shows mismatch between the output and the ground truth (frames selected by users not present in the output) and those shown using blue dotted line show frames present in the output and not present in the ground truth. The overlapping of plot generated by the human summaries and the proposed method shows that the summaries generated by the proposed method is similar to that of the human-generated summaries corresponding to each type of videos.

IV. CONCLUSION

Domain-independent dynamic video summarization is gaining interest in research community due to the massive growth of videos. In this paper, we propose domain independent video summarization system based on the combination of high and low-level features. The results show that the proposed method attain good results compared to the other state-of-the-art techniques. Future work includes fine-tuning layers of SAE to extract more representative features from video frames so that summarization results can be improved. Exploitation of information from spatial as well as temporal dimensions gives good recognition accuracy.

ACKNOWLEDGMENT

The authors would like to thank University of Kerala, Thiruvananthapuram, India, for providing financial support for this work under University Junior Research Fellowship Scheme.

REFERENCES

[1] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini, "STIMO: Still and moving video storyboard for the Web scenario," *Multimedia Tools Appl.*, vol. 46, no. 1, pp. 47–69, 2010.

[2] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 505–520.

[3] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 3, no. 1, 2007, Art. no. 3.

[4] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, "A fuzzy video content representation for video summarization and content-based retrieval," *Signal Process.*, vol. 80, no. 6, pp. 1049–1067, 2000.

[5] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using delaunay clustering," *Int. J. Digit. Libraries*, vol. 6, no. 2, pp. 219–232, 2006.

[6] S. E. F. de Avila, A. da Luz, Jr., A. de A. Araújo, and M. Cord, "VSUMM: An approach for automatic video summarization and quantitative evaluation," in *Proc. 21st Brazilian Symp. Comput. Graph. Image Process. (SIBGRAPI)*, Oct. 2008, pp. 103–110.

[7] G. Guan, Z. Wang, S. Lu, J. D. Deng, and D. D. Feng, "Keypoint-based keyframe selection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 4, pp. 729–734, Apr. 2013.

[8] T. Hu, Z. Li, W. Su, X. Mu, and J. Tang, "Unsupervised video summaries using multiple features and image quality," in *Proc. IEEE 3rd Int. Conf. Multimedia Big Data (BigMM)*, Apr. 2017, pp. 117–120.

[9] T. Hu and Z. Li, "Video summarization via exploring the global and local importance," *Multimedia Tools Appl.*, vol. 77, no. 17, pp. 22083–22098, 2018.

[10] B. Zhao and E. P. Xing, "Quasi real-time summarization for consumer videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2513–2520.

[11] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang, "A generic framework of user attention model and its application in video summarization," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 907–919, Oct. 2005.

[12] M. Fei, W. Jiang, and W. Mao, "Creating memorable video summaries that satisfy the user's intention for taking the videos," *Neurocomputing*, vol. 275, pp. 1911–1920, Jan. 2018.

[13] N. Shroff, P. Turaga, and R. Chellappa, "Video Précis: Highlighting diverse aspects of videos," *IEEE Trans. Multimedia*, vol. 12, no. 8, pp. 853–868, Dec. 2010.

[14] M. B. Fayk, H. A. El Nemr, and M. M. Moussa, "Particle swarm optimisation based video abstraction," *J. Adv. Res.*, vol. 1, no. 2, pp. 163–167, 2010.

[15] M. Gygli, H. Grabner, and L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3090–3098.

[16] Y. J. Lee and K. Grauman, "Predicting important objects for egocentric video summarization," *Int. J. Comput. Vis.*, vol. 114, no. 1, pp. 38–55, 2015.

[17] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial LSTM networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2017, pp. 1–10.

[18] T. Yao, T. Mei, and Y. Rui, "Highlight detection with pairwise deep ranking for first-person video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 982–990.

[19] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 766–782.

[20] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with domain knowledge and hidden markov models," *Pattern Recognit. Lett.*, vol. 25, no. 7, pp. 767–775, 2004.

[21] H. Sundaram, L. Xie, and S.-F. Chang, "A utility framework for the automatic generation of audio-visual skims," in *Proc. 10th ACM Int. Conf. Multimedia*, 2002, pp. 189–198.

[22] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1346–1353.

[23] D. Liu, G. Hua, and T. Chen, "A hierarchical visual model for video object summarization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2178–2190, Dec. 2010.

[24] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg, "Webcam synopsis: Peeking around the world," in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.

[25] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a long video short: Dynamic video synopsis," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2006, pp. 435–441.

[26] Z. Chen, G. Lv, L. Lv, T. Fan, and H. Wang, "Spectrum analysis-based traffic video synopsis," *J. Signal Process. Syst.*, vol. 90, nos. 8–9, pp. 1257–1267, 2018.

[27] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 13–26.

[28] J. Päivärinta, E. Rahtu, and J. Heikkilä, "Volume local phase quantization for blur-insensitive dynamic texture classification," in *Proc. Scandin. Conf. Image Anal.* Berlin, Germany: Springer, 2011, pp. 360–369.

[29] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila, "Recognition of blurred faces using local phase quantization," in *Proc. 19th Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2008, pp. 1–4.

[30] B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, vol. 8206. Springer, 2013, pp. 611–618.

[31] Z. Guo, L. Gao, X. Zhen, F. Zou, F. Shen, and K. Zheng, "Spatial and temporal scoring for egocentric video summarization," *Neurocomputing*, vol. 208, no. 5, pp. 299–308, 2016.

[32] V. Zocca, G. Spacagna, D. Slater, and P. Roelants, *Python Deep Learning*. Birmingham, U.K.: Packt publishing, 2017.

[33] C. Gravelines, "Deep learning via stacked sparse autoencoders for automated voxel-wise brain parcellation based on functional connectivity," Ph.D. dissertation, Dept. Comput. Sci., Univ. Western Ontario, London, U.K., 2014, pp. 1–75.

[34] E. Hosseini-Asl, J. M. Zurada, and O. Nasraoui, "Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2486–2498, Dec. 2016.

[35] N. Ejaz, I. Mehmood, and S. W. Baik, "Efficient visual attention based framework for extracting key frames from videos," *Signal Process., Image Commun.*, vol. 28, no. 1, pp. 34–44, Jan. 2013.

JESNA MOHAN received the M. Tech. degree in computer science with the specialization in digital image computing from the University of Kerala, India. She is currently a Research Scholar at the Department of Computer Science, University of Kerala. Her research interests include image processing, video analysis, and pattern recognition.

MADHU S. NAIR (SM'15) received the bachelor's degree (B.C.A.) (Hons.) in computer applications and the master's degree (M.C.A.) (Hons.) in computer applications from Mahatma Gandhi University in 2000 and 2003, respectively, the master's degree in technology (M.Tech.) (Hons.) in computer science (with specialization in digital image computing) from the University of Kerala in 2008, and the Ph.D. degree in computer science (image processing) from Mahatma Gandhi University in 2013. He has qualified the National Eligibility Test for Lectureship conducted by the University Grants Commission in 2004 and the Graduate Aptitude Test in Engineering conducted by IIT in 2006. He has published around 69 research papers in reputed International Journals and Conference Proceedings published by the IEEE, Springer, Elsevier, Wiley, and IOS Press. His research interests include digital image processing, pattern recognition, computer vision, data compression, and soft computing. He is a member of Association for Computing Machinery and the International Association of Engineers and an Associate Life Member of the Computer Society of India (CSI). He was a recipient of prestigious national awards, such as the AICTE Travel Grant Award 2008, the INAE Innovative Student Projects Award 2009 for Best M.Tech. Thesis, the CSI Paper Presenter Award at International Conference, the AICTE Career Award for Young Teachers (CAYT), the IEI Young Engineers Award from 2013 to 2014, and the SSI Young Systems Scientist Award 2015. He has also received the Most Active Editorial Board Member/Reviewer Award from the *International Arab Journal of Information Technology* (IAJIT), and the Best Paper Award during International Conference on Advances in Computing and Communications (ACC 2011). He is serving as a reviewer for around 38 International Journals published by the IEEE, Elsevier, Springer, Wiley, and World-Scientific, and also served as a Technical Programme Committee Member/Reviewer for several reputed International Conferences. He is currently an Associate Editor of the prestigious IEEE Access Journal and also the Editorial Board Member of the IAJIT.