IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 June 2022, accepted 30 June 2022, date of publication 5 July 2022, date of current version 13 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3188638

RESEARCH ARTICLE

CNN and HEVC Video Coding Features for Static Video Summarization

OBADA ISSA AND TAMER SHANABLEH, (Senior Member, IEEE)
Department of Computer Science and Engineering, American University of Sharjah, Sharjah, United Arab Emirates

Corresponding author: Obada Issa (b00071518@aus.edu)

This work was supported in part by the Open Access Program through the American University of Sharjah under Grant FRG22-E-E44.

ABSTRACT This study proposes a novel solution for the detection of keyframes for static video summarization. We preprocessed the well-known video datasets by coding them using the HEVC video coding standard. During coding, 64 proposed features were generated from the coder for each frame. Additionally, we converted the original YUVs of the raw videos into RGB images and fed them into pretrained CNN networks for feature extraction. These include GoogleNet, AlexNet, Inception-ResNet-v2, and VGG16. The modified datasets are made publicly available to the research community. Before detecting keyframes in a video, it is important to identify and eliminate duplicate or similar video frames. A subset of the proposed HEVC feature set was used to identify these frames and eliminate them from the video. We also propose an elimination solution based on the sum of the absolute differences between a frame and its motion-compensated predecessor. The proposed solutions are compared with existing works based on an SIFT flow algorithm that uses CNN features. Subsequently, an optional dimensionality reduction based on stepwise regression was applied to the feature vectors prior to detecting key frames. The proposed solution is compared with existing studies that use sparse autoencoders with CNN features for dimensionality reduction. The accuracy of the proposed key-frame detection system was assessed using the positive predictive values, sensitivity, and F-scores. Combining the proposed solution with Multi-CNN features and using a random forest classifier, it was shown that the proposed solution achieved an average F-score of 0.98.

INDEX TERMS Convolution neural network, duplicate frames, sparse auto encoders, static video summarization, video coding, high efficiency video codec (HEVC).

I. INTRODUCTION

With the surge of the Internet and surveillance footage, there is a vast number of digital videos. The need to summarize these videos in databases is crucial. This is where video summarization is useful. Video summarization is the process of generating a meaningful summary of the original video, which, in turn, facilitates video retrieval, anomaly detection, and activity monitoring [1]. Video summarization can be achieved by using several methods or techniques. These techniques can be categorized into two groups [2]. The first is selecting sections or shortcuts of the original video, and the second is selecting key frames that represent the original video. Key frame selection techniques require manual human annotation of videos to automate the training process of frame selection. This technique is the most common among the two aforementioned video summarization techniques. Therefore, this study focuses on video summarization by automatically selecting key frames in a video.

Video summarization is computationally demanding, and more efficient approaches are required. If all frames within a video are examined for selection, the summarization process can be slow, and time and computational power are wasted on redundant or similar frames. Additionally, for any set of features, space reduction should be used to accelerate the process and ensure that only meaningful features are considered [3]. This study aims to address these two issues. Deep learning has gained popularity in recent years for generation tasks in image and video processing. Many tools can be used either independently or in combination to achieve the desired results. Most notably, conventional neural networks (CNN) [4] and random forests (RFs) [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

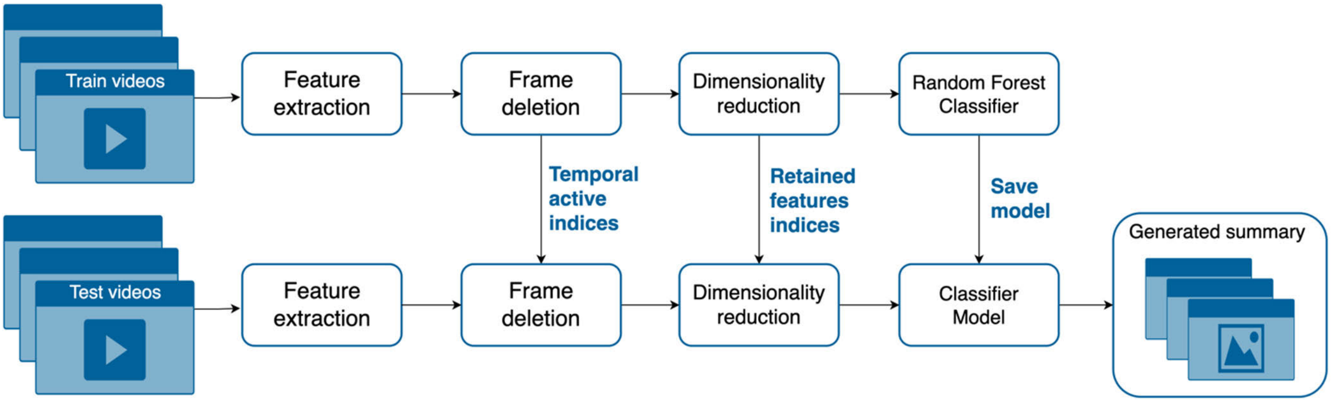


FIGURE 1. General overview of the system architecture. Feature extraction is done either through CNNs or a custom HEVC decoder. Dimensionality reduction using Stepwise regression. Frame elimination using S.A.D of motion estimation and compensation or HEVC features.

The deep learning community falls in the computer and data science fields, while video compression communities fall in the field of electrical engineering. The separation in these research areas often leads the deep learning community to lack sufficient comprehension of video compression. In the deep learning field, with the growth of the High Efficiency Video Codec (HEVC) [6] video standard, HEVC information found in the video bitstream is often ignored and not used to its full potential. This research intends to use low-level HEVC features in combination with CNN features to aid in video summarization. The integration of HEVC features can take several forms, including merging bit stream information with CNN features derived from frames or constructing a separate channel with HEVC-based bit stream information and then fusing the output with CNN-based results.

This study also introduces a novel method for reducing the dimension of the feature space obtained from CNNs using stepwise regression, along with a novel method for similar frame elimination based on HEVC features. A general overview of the system architecture is in Fig. 1.

A summary of our research contributions to the body of the literature is in the following points:

- We introduce low-level HEVC feature sets suitable for video summarization.
- We propose a novel method for frame elimination based on motion estimation and compensation.
- We propose a novel method for frame elimination based on HEVC features.
- These contributions resulted in promising results that outperform state-of-the-art works in the literature.

This paper is structured as follows: Section II is the literature review, Section III presents the datasets and data preprocessing, Section IV proposes the low-level HEVC feature set, Section V discusses elimination of similar or redundant frames, Sections V-A and V-B are frame elimination based on motion estimation and compensation and based on HEVC features, respectively. Section VI discusses dimensionality reduction of the feature space, Section VI-A is about the Sparse Autoencoder, Section VI-B proposes dimensionality reduction of the feature space using stepwise regression, Section VII contains the experimental results. Section VII-A explains the evaluation criteria, followed by Section VII-B with the experimental results, then Section VII-C with final discussion and Section VIII presents the conclusion and future work.

II. LITERATURE REVIEW

Video summarization has been researched extensively in the past decade owing to the incredible surge of online videos and the challenges that come with sorting and saving these videos in huge databases. This section summarizes the efforts found in the literature that tackle the video summarization task, especially deep learning-based methods.

A. LSTM-BASED

When casting the task as a structured prediction problem, [7] used Long Short-Term Memory (LSTM) to model the variable-range temporal dependency among video frames to derive both representative and compact video summaries. In [8], a 2-layer network with two bidirectional LSTMs is proposed, that acts as filters to identify shot boundaries to capture forward and backward temporal dependencies among shots to predict which shots are most representative of the video content. A stacked memory network called SMN with LSTM layers was presented in [9] to model the long dependency among video frames to reduce redundancy in the produced summaries. Multi-video summarization (MVS) was achieved by [10] with a two-tier framework that performs target-appearance-based shot segmentation, extracts deep features from frames, and passes them to a bidirectional LSTM to acquire probabilities and generate a summary. An attentive encoder–decoder network was proposed in [11], where the encoder is a bidirectional LSTM to encode the contextual information among frames, and the decoder is two attention-based LSTM networks that use additive and multiplicative objective functions. A self-attention binary neural tree (SABT-Net) model is proposed by [12] that uses GoogleNet network, shot encoding, branch routing, self-attention, and score prediction modules to form the summaries. A TTH-RNN was proposed in [13] which contains a tensor-train embedding layer and a hierarchical LSTM that captures intra-subshot temporal dependence and encodes inter-subshot temporal dependence from the forward and backward directions to determine the importance of every frame and form the summaries.

B. CNN-BASED

The researchers in [14] designed a deep neural network with a clustering-based summarization technique that maps videos and descriptions to a common semantic space and trained it with labeled videos and their descriptions. As proposed in [15], variance loss is introduced to allow a network to predict the output scores for each frame. They also designed a two-stream network (CSNet) that utilized local (chunk) and global (stride) temporal views. In [16], the authors proposed fully convolutional sequence models, where they established a connection between semantic segmentation and video summarization. An unsupervised deep summarization network (DSN) was developed by [17] which predicts a selection probability and selects frames based on probability distributions to form video summaries. The authors in [18] leveraged spatiotemporal learning with three-dimensional convolutional neural networks (3D-CNN), LSTMs, and recurrent neural networks (RNN) to detect soccer video highlights. The framework in [19] is a two-stream deep architecture with cost-sensitive learning. A spatial stream uses RGB images to represent the appearance of frames, and a temporal stream uses motion vectors to represent and extract the temporal information of the input video. An encoder-decoder CNN structure was adopted by [20] where the encoder is a diagnostic view plane detection network, and the decoder feeds feature maps into a bi-directional LSTM to analyze features of past and future frames. The final reinforcement learning (RL) network selects key frames for the summary. Video summarization was performed by [21] on the Internet of Things (IoT) surveillance domain by designing a CNN framework that performs shot segmentation and image memorability, and aesthetic and entropy features are used to maintain the diversity of the summary. The authors in [22] used a sparse autoencoder that combines feature vectors derived from four famous image CNNs into a reduced space and a random forest classifier to select key frames.

C. GAN-BASED

The authors in [23] specified a novel generative adversarial framework (GAN) with an LSTM summarizer autoencoder for selecting frames and an LSTM discriminator to distinguish between the original video and its reconstruction from the summarizer. A GAN framework was developed by [24] where the generator was an attention-aware Ptr-Net, and the discriminator was a 3D-CNN classifier. The researchers in [25] presented an unsupervised GAN with an attention mechanism to identify significant parts of a video.

D. OTHER MODELS

A "retrospective encoder" is used in [26] that embeds the predicted summary into an abstract semantic space and compares it to the embedding of the original video in the same space and try to minimize the distance between these two spaces. HEVC intra-frame coding was leveraged by [27] by fusing weighted luminance and chrominance values along with texture feature vectors, and then applying thresholding to select frames for summarization. In [28], motion information between frames is used, where spatio-temporal information is extracted, an inter-frame motion is generated from it, and a self-attention model selects key frames for summarization. Motion-based frame selection and a novel clustering validity index were used in [29] to segment shots and select candidate frames by evaluating their forward and backward motion, where the frame with the largest motion was taken in each segmentation to form the video summary.

III. DATASETS AND DATA PREPROCESSING

In this study, we assessed the performance of the proposed solutions using two benchmark datasets for static video summarization, namely, VSUMM and OVP. The datasets include CNN features generated using GoogleNet and ground truth data. The datasets and ground-truth data are publicly available in [30].

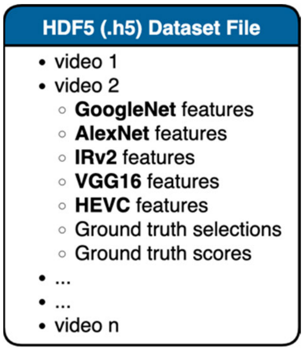
The VSUMM dataset contains 50 videos from websites such as YouTube containing several genres (cartoons, news, sports, commercials, tv-shows, and home videos) and have a duration of 1–10 min at 30 fps. The OVP (Open Video Project) dataset has 50 videos from Open Video Project in MPEG-1 format at 30 fps. The videos were distributed among several genres (documentary, educational, ephemeral, historical, and lecture) and have a duration of 1-4 minutes.

In our data preparation process, we retrieved the original videos from [30] and converted them into RGB images. These images were then fed into a number of pretrained CNN networks for feature extraction. The CNN networks used for feature extraction were AlexNet [31], Inception-ResNet-v2 (IRv2) [32], and VGG16 [33]. GoogleNet features for both the OVP and VSUMM datasets were acquired online from a public HDF5 dataset from [30]. Each of the CNN networks has it unique input size, and therefore, our input frames were resized prior to feature extraction to match the respective input size of each of the CNNs. The resulting feature space from each network along with the input sizes used are all summarized in Table 1.

TABLE 1. Input and feature vector sizes for networks used for feature extraction.

Network	GoogleNet	AlexNet	IRv2	VGG16	HEVC
Input size	224x224	227x227	299x299	224x224	Original
Features	1024	4096	1536	4096	64

In addition, we converted the original videos into YUV images and then to an HEVC/H.265 video coder [34]. We modified the coder to generate low-level features, as described in the next section. HEVC features have been successfully used in many applications, including encoding speedup [35], video transcoding [35], data embedding [36], double and triple compression detection [37] and saliency detection [38]. Eventually, all the feature variables are added to the .h5 files of the OVP and VSUMM, as illustrated in Fig. 2. In this work, we tested the proposed solution using HEVC features, CNN features, and a combination of both.



HDF5 (.h5) Dataset File

- video 1
- video 2
  - **GoogleNet** features
  - **AlexNet** features
  - **IRv2** features
  - **VGG16** features
  - **HEVC** features
  - Ground truth selections
  - Ground truth scores
- ...
- ...
- video n

FIGURE 2. The structure of the HDF5 file with CNN and HEVC features.

IV. PROPOSED LOW-LEVEL HEVC FEATURE SET

As mentioned in Section III, the HEVC codec was used to compress the videos. Thus, HEVC can be used to extract rich feature sets based on quadratic recursive splitting of coding units (CUs).

TABLE 2. Proposed HEVC feature set per frame. A custom HEVC decoder was used to extract these features.

	Feature ID	Feature variable
Averaged per frame	**1**	Number of CU parts
	2	MVD bits per CU
	3	CU bits excluding MVD bits
	4	Percentage of intra CU parts
	5	Percentage of skipped CU parts
	6	Number of CUs with depth 0 (i.e 64x64)
	7	Number of parts with depth 1 (i.e 32x32)
	8	Number of CUs with depth 2 (i.e 16x16)
	9	Number of parts with depth 3 (i.e 8x8)
	10-18	Standard deviation of feature IDs 1-9 per frame
	19	Max CU depth per frame
	20	For CUs with depth > 0, $\log_2(\lvert sum\ of\ MVD \rvert)$?
	21	For CUs with depth = 0, $\log_2(\lvert sum\ of\ MVD \rvert)$?
Averaged per frame	**22**	Row-wise SAD of the CU prediction error
	23	Column-wise SAD of the CU prediction error
	24	Ratio of gradients (i.e feature 22 divided by feature 23) per CU
	25	Total distortion per CU as computed by the HEVC encoder
	26-29	Standard deviation of feature IDs 22-25 per frame
	30	Per frame: Summation of variance of the x and y components of all MVs
	31-47	Histogram of x-component of all MVs per frame (using 16 pins)
	48-64	Histogram of y-component of all MVs per frame (using 16 pins)

CUs in HEVC can have different depths ranging from 0, which corresponds to a block of 64 × 64 pixels, to a depth of 3, which corresponds to a block of 8 × 8 pixels. A CU can be split into four square parts, and each part can be recursively split into subparts. Splitting occurs following a rate distortion criteria which considers the spatio-temporal activities of the underlying CU.

In this study, we proposed a set of 64 feature variables per frame for video summarization. The selection of variables quantifies the spatiotemporal activities of the video frames.

The variables are listed in Table 2. In the table, MVD is short for motion vector difference, SAD is short of the sum of absolute differences, and CU is short for the coding unit.

The modified OVP and SUMME datasets with the proposed HEVC features and Multi-CNN features are available at https://bit.ly/HEVC-SVS.

V. ELIMINATION OF SIMILAR FRAMES

In [22] it was mentioned that redundant frames increase the complexity of detecting keyframes. A redundant frame is defined as one that is identical or very similar to the previous frame. In the following subsections, we propose two solutions to quantify frame redundancy based on the introduction of a temporal activity index. In the first proposed solution, such an index is computed using low-level HEVC feature variables. In the second proposed solution, the index is computed based on motion compensation and motion vector variance.

Clearly varying the framerate can affect the percentage of eliminated video frames. More specifically, with higher frame rates, the similarity between consecutive frames becomes higher and hence higher percentage of frames are eliminated. However, since the work reported in the literature all use the same datasets with the same framerates, we will discuss the effect of framerate on frame elimination in future work.

In [22] this approach was used for elimination, which is based on the work reported in [39]. Briefly, each frame in a video is converted to a SIFT image, where each pixel is represented by a 128-D SIFT descriptor. The flow vectors were then calculated between consecutive SIFT images. The global displacement vector of a given video frame was calculated by adding all magnitudes of the flow vectors per frame. Eventually, redundant frames are identified based on local thresholding of the magnitude of the global vectors.

A. PROPOSED ELIMINATION OF SIMILAR OR FRAMES BASED ON HEVC FEATURES

In the proposed list of HEVC feature variables, there exists a set of variables that indicates the low temporal activity of video frames. Such low temporal activity indicates that the current frame is similar to its previous frames, and therefore, can be eliminated. In this solution, we use the sum of the following HEVC feature variables to produce a temporal activity index: the lower the index, the lower the temporal activity. The feature variables are listed in Table 3.

TABLE 3. List of HEVC feature subset variables used to produce a temporal activity index.

Feature ID	Feature variable (averaged over a frame)
1	Number of CU parts
2	MVD bits per CU
3	CU bits excluding MVD bits
5	Percentage of skipped CU parts
6	Number of CUs with depth 0 (i.e 64x64)
7	Number of parts with depth 1 (i.e 32x32)
22	Row-wise SAD of the CU prediction error
23	Column-wise SAD of the CU prediction error
24	Ratio of gradients (i.e feature 22 divided by feature 23) per CU
25	Total distortion per CU as computed by the HEVC encoder
30	Summation of variance of the x and y components of all MVs

In the implementation process, the video dataset was split into training and test data, as explained in the experimental results section. For each split of the dataset, we used video frames with ground truth zero in the training data to determine the average value of the features listed in Table 3. Video frames with ground truth zero are those that are not part of the video summary. The averages were then summed to compute the temporal activity index. To vary the value of the mentioned index, the summation of the standard deviations of the feature variables can be added, as illustrated in Equation 1.

$$TAI = \sum\nolimits_{i=1}^{I} \bar{f_i} + c \sum\nolimits_{i=1}^{I} \sigma_{f_i} \tag{1}$$

where TAI is the temporal activity index, f_i is feature i from Table 2, and i ranges from 1 to 11, which is the total number of features used. Constant c ranges from 0 to 1 and can be used to vary the value of the temporal activity index. In this work, it is set to 0.35 using empirical testing. Consequently, a video frame from the test data was eliminated according to the Boolean condition presented in Equation 2.

$$Eliminate\ frame_j = \begin{cases} False, & if\ \sum\nolimits_{i=1}^{I} \bar{f}_{i,j} > TAI \\ True, & if\ \sum\nolimits_{i=1}^{I} \bar{f}_{i,j} \leq TAI \end{cases} \tag{2}$$

where the frame at index j belongs to the test video set and $f_{i,j}$ is feature i from Table 3 of the test frame at index j.

B. PROPOSED ELIMINATION OF SIMILAR FRAMES BASED ON MOTION ESTIMATION AND COMPENSATION

The temporal activity index is calculated based on the sum of the absolute differences between a video frame and its preceding frame after applying motion compensation. In the implementation, we used the optical flow to compute the motion between two consecutive frames at indices j and j-1. This results in a list of motion vectors, one for each pixel in frame j. These motion vectors are then used to motion compensate $frame_{j-1}$, according to Equation 3:

$$MC\_frame_{j-1}(r, c) = frame_{j-1}(x - Vx_{r,c} - Vy_{r,c}) \tag{3}$$

where MC is short for motion compensation, r and c are the pixel coordinates in a video frame, and $Vy_{r,c}$ and $Vy_{r,c}$ are the x and y components of the motion vector belonging to the pixel at indexes r and c, respectively.

The sum of the absolute differences (SAD) between $frame_j$ and its previous motion-compensated frame (i.e., MC_frame_{j-1}) is then computed according to Equation 4:

$$SAD_j = \sum\nolimits_r \sum\nolimits_c \left| frame_j(r, c) - MC\_frame_{j-1}(r, c) \right| \tag{4}$$

The higher the SAD, the higher the temporal activity between the two frames. To further quantify the temporal activity, we propose weighing the SAD using the variance of the motion vectors. This is because a high variance in motion vectors is another indicator of high temporal activity. Therefore, Equation 4 can be expanded as follows:

$$SAD_j = \sqrt{\sigma_{Vx} + \sigma_{Vy}} * \sum_r \sum_c \left| frame_j(r, c) - MC\_frame_{j-1}(r, c) \right| \tag{5}$$

Consequently, a frame from the test data is eliminated if it's SAD is less than the P^{th} percentile of all the sums of the absolute differences of a test dataset. Formally, a frame was eliminated according to the following Boolean condition.

$$Eliminate\ frame_j = \begin{cases} False, & if\ SAD_j \geq SAD_{\lceil \frac{P}{100} * J \rceil} \\ True, & if\ SAD_j < SAD_{\lceil \frac{P}{100} * J \rceil} \end{cases} \tag{6}$$

where J is the total number of frames in the video test set minus one (as the motion estimation starts from the second frame). The subscript of SAD on the right-hand side indicates the percentile rank.

VI. DIMENSIONALITY REDUCTION

As mentioned previously, in this study, we propose the use of HEVC features for video summarization, and we also use CNN features generated from the obtained networks, as proposed by [22]. This will allow us to compare our work with existing solutions and it will allow us to test the suitability of combining the proposed HEVC features with CNN features.

The CNN features were generated using IRv2, AlexNet, and VGG16. While GoogleNet features were acquired from [29], as mentioned in Section III. When all four CNN features were combined, the total length of the feature vector per video frame was 10,752. In [22], a sparse autoencoder (SAE) was used to reduce the dimensionality of the feature space to 500 feature variables. Reduction is performed to ensure that the best encoded representation of the feature vectors is chosen and to ease the training process.

A. DIMENSIONALITY REDUCTION BASED ON SAE

The SAE architecture includes an input layer that takes in the entirety of the 10,752-feature vector for a given frame with a node representing each feature, a latent layer with a reduced feature space, and an output layer with a feature vector of length 500. The SAE works similarly to a traditional autoencoder except that it adds a sparsity penalty. Every feature was compared against a weight decay penalty, which was one thousand the value of the sparsity penalty. This means that if the autoencoder deems that the feature is not as useful as the other, it does not make it into the reduced space.

In this study, in addition to the use of SAEs, we propose the use of stepwise regression for dimensionality reduction.

B. PROPOSED STEPWISE REGRESSION SOLUTION

Stepwise regression is a supervised predictor-selection algorithm in which the choice of predictive variables is performed automatically [40]. The use of stepwise regression in video-based intelligent systems was first proposed by the authors in [41]. Since then, it was successfully used with video codec as reported in [38], [42] and [43] to mention a few.

In this work, we propose the use of stepwise regression to reduce the dimensionality of both HEVC and CNNs features, in which we treat the feature variables as predictors and the class labels as response variables. Because stepwise regression is a supervised predictor selection algorithm, it is important to implement it on the training data only. Consequently, the indices of the retained feature variables are stored and used to reduce the dimensionality of the test data, as illustrated in Fig. 3. For completeness, we provide a summary of the stepwise regression algorithm.

Dimensionality reduction methods are usually forward, starting with one feature and adding features to reach the optimal model, or backward, starting with all features and dropping one feature at a time to reach the optimal model. Stepwise, it combines the forward and backward methods, meaning that at each iteration, a feature can be dropped or added.

For a set of features $x_1, x_2, \ldots, x_k$. F_{in} is the F-random feature for the feature to be added to the model, whereas F_{out} is the feature to be dropped from the model. The steps for stepwise regression are as follows:

1- Form 1-itemsets from all features to produce single-feature models:

$$h\,(x) = \theta_0 + \theta_1 x_1 \tag{7}$$

where $h(x)$ is the hypothesis that the added features are needed for the classification task. x_1 was one of the features that yielded the highest F-score. f_1 is the statistic of x_1 and is given by the following formula:

$$f_1 = \frac{SS_R(\theta_2|\theta_1\theta_0)}{MS_E(x_2, x_1)} \tag{8}$$

where SS_R is the regression sum square error and MS_E is the mean square error.

2- From the equation obtained above for the 1-itemset feature, we examined the rest of the $k-1$ features that can, when combined with $h(x)$, produce a higher hypothesis than $h(x)$ by itself. We add x_2 if its f_2 is greater than F_{in} and obtain the following:

$$f_2 = \frac{SS_R(\theta_1|\theta_2\theta_0)}{MS_E(x_1, x_2)} \tag{9}$$

$$h\,(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \tag{10}$$

After adding x_2, we check whether x_1 needs to be removed by comparing f_1 to the new F_{out}. If f_1 is lesser, x_1 decreases.

3- The remaining $k-2$ features are examined to obtain x_3 and get the following hypothesis:

$$h\,(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \tag{11}$$

The algorithm continues until there are no features to add or drop.

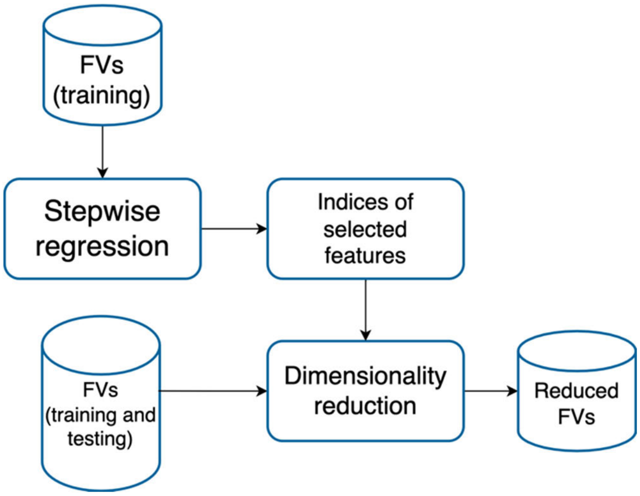


FIGURE 3. General overview of how Stepwise regression is used for feature space dimensionality reduction. SW retains the indices of best features so they can be used later for testing on new data.

VII. EXPERIMENTAL RESULTS

A. EVALUATION CRITERIA

The used quantitative performance metrics are:

- Positive predictive values (PPV): Percentage of true positive predictions over all positive predictions. Where P_T is the true positive prediction, and P is all positive predictions.

$$\boldsymbol{PPV = P_T/P} \tag{12}$$

- Sensitivity (S): Percentage of true positive predictions over the users' ground truth (at the same frame indices). where P_T is the true positive prediction and P_u is the user-selected frames and their indices.

$$\boldsymbol{S = P_T/P_u} \tag{13}$$

- F-score: F-measure (or F-score) is the harmonic mean of the precision and recall scores. It provides a combined view when either of these scores is not sufficient to describe the imbalanced classification problem, such as the frame selection problem, in which most frames are not selected and only a few are selected.

$$F - score = 2 \times PPV \times S/PPV + S \qquad (14)$$

B. EVALUATION

After data preprocessing and preparation, we proceeded with experimenting our proposed methods. The proposed dimensionality reduction method using stepwise regression was compared with the SAE in [22]. In addition, the proposed elimination of similar frame methods using ME+MC (Section V-B.) and low-level HEVC features (Section V-A), respectively, were compared against the SIFT flow algorithm used in [22]. Finally, the proposed HEVC feature set was tested against the features extracted from well-known CNN models.

The trial runs reported have the following set-up: Each run consists of 5 non-overlapping folds. In other words, the data (videos) are split in an 80%-20% fashion for training and testing, respectively. For single CNN runs, the feature vector lengths are mentioned in Section-III. For Multi-CNN runs, the feature vector length is 10,752, which is a combination of the 4 CNNs (GoogleNet: 1024 features, AlexNet: 4096, IRv2: 1536 features, VGG16: 4096 features). For Multi-CNN & HEVC runs, the feature vector length is 10,816, which is a combination of the 4 CNNs and the HEVC feature set (GoogleNet: 1024 features, AlexNet: 4096, IRv2: 1536 features, VGG16: 4096 features, HEVC: 64). The results reported for each trial run are the average results for of the 5 testing folds. The metrics reported are PPV, Sensitivity and F-score, which correspond to equations 12, 13 and 14, respectively. Further evaluations like the confusion matrices and the models' run times can be found in the appendix section at the end of this paper. The experiments were conducted on a PC provided by the American University of Sharjah with Intel i7 (7th gen) CPU, 16 GB of RAM and NVIDIA GTX 1070 GPU.

1) OVP DATASET – SINGLE-CNN RESULTS

The following are the experimental results for the OVP dataset using a random forest classifier with features derived from GoogleNet (1024 features), AlexNet (4096 features), IRv2 (1536 features), and VGG16 (4096 features). We also present the results from using the proposed HEVC feature set (64 features).

In Table 4, the accuracy of the key-frame detection using different methods is presented. The best detection results are obtained when using the proposed HEVC features combined with stepwise regression and HEVC feature-based frame elimination (Section V-B). For features generated from the pertained CNNs, we experimented with dimensionality reduction using the existing work of [22], which is based on SAEs, and using the proposed stepwise regression solution. Higher detection accuracies have been reported for the latter.

TABLE 4. Experimental results on the OVP dataset - Single-CNN.

Feature Sets	Feature Reduction	Frame Elimination	Metrics		
			PPV	S	Fs
HEVC (Proposed)	None	SIFT	0.59	0.71	0.83
		ME+MC	0.71	0.85	0.92
		HEVC	**0.87**	**0.86**	**0.93**
GoogleNet	SAE [22]	SIFT [22]	0.50	0.93	0.61
	SW	SIFT	0.55	0.91	0.86
		ME+MC	0.63	0.95	0.87
		HEVC	**0.59**	**0.96**	**0.88**
AlexNet	SAE [22]	SIFT [22]	0.70	0.89	0.78
	SW	SIFT	0.54	0.93	0.86
		ME+MC	0.62	0.95	0.87
		HEVC	**0.59**	**0.97**	**0.88**
IRv2	SAE [22]	SIFT [22]	0.75	0.84	0.79
	SW	SIFT	0.54	0.93	0.86
		ME+MC	**0.63**	**0.96**	**0.88**
		HEVC	0.59	0.94	0.87
VGG16	SAE [22]	SIFT [22]	0.68	0.61	0.64
	SW	SIFT	0.54	0.71	0.85
		ME+MC	**0.62**	**0.74**	**0.87**
		HEVC	0.59	0.76	0.87

The results in the table also indicate that eliminating replicated frames based on HEVC features results in a higher detection accuracy compared to the use of the SIFT-based algorithm. The detection results for features based on IRv2 and VGG16 peaked with the use of the proposed ME+MC frame elimination solution described in Section V-B.

2) OVP DATASET – MULTI-CNN RESULTS

In Table 5, we present the detection results for the OVP dataset after combining all features. In one experiment, all CNN features were combined (Multi-CNN), and in another experiment, we combined the proposed HEVC features with all CNN features (Multi-CNN and HEVC).

The Multi-CNN model performed better than all previous models with single CNN features by up to a 6% increase in performance across all metrics. This shows how the stepwise regressor can retain the best features from multiple CNNs and attain satisfactory results. The detection accuracy resulting from Multi-CNN surpasses the use of individual CNNs, as reported in Table 4. More noticeably, the use of Multi-CNN and HEVC resulted in accuracy of 0.98 F-score.

TABLE 5. Experimental results on the OVP dataset - Multi-CNN.

Feature Sets	Feature Reduction	Frame Elimination	Metrics		
			PPV	S	Fs
Multi-CNN	SAE [22]	SIFT [22]	0.78	0.86	0.82
	SW	SIFT	0.55	0.94	0.87
		ME+MC	0.62	0.96	0.90
		HEVC	**0.59**	**0.97**	**0.93**
Multi-CNN & HEVC (proposed)	SW	SIFT	0.54	0.93	0.96
		ME+MC	0.65	0.95	0.97
		HEVC	**0.83**	**0.96**	**0.98**

3) OVP DATASET – VS. EXISTING WORKS

Figure 4 shows our best model on the OVP dataset, with Multi-CNN and HEVC feature sets and HEVC-based frame elimination against the state-of-the-art [22], VISCOM [44], VRHDPS [45] and VSUMM [46]. The results obtained from our proposed solution for the OVP dataset surpassed those of the existing work across all metrics.

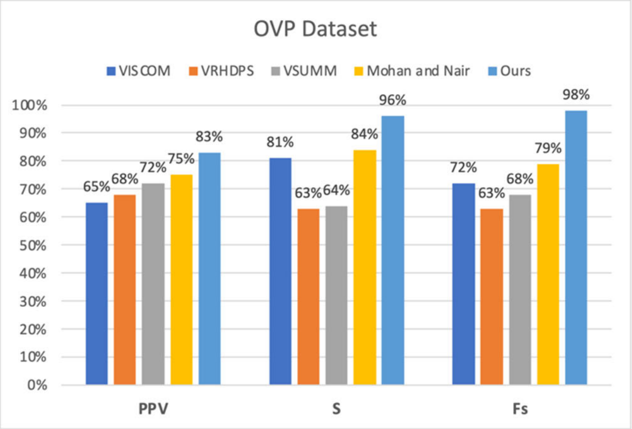


FIGURE 4. This graph shows our top performing model (Multi-CNN & HEVC and HEVC-based frame elimination) in terms of F-score on the OVP dataset compared with existing works in the literature.

TABLE 6. Experimental results on the VSUMM dataset - Single-CNN.

Feature Sets	Feature Reduction	Frame Elimination	Metrics		
			PPV	S	Fs
HEVC (proposed)	None	SIFT	0.54	0.52	0.51
		ME+MC	0.71	0.82	0.75
		HEVC	**0.88**	**0.86**	**0.86**
GoogleNet	SAE [22]	SIFT [22]	0.61	0.81	0.69
	SW	SIFT	0.56	0.82	0.76
		ME+MC	0.70	0.82	0.76
		HEVC	**0.68**	**0.84**	**0.78**
AlexNet	SAE [22]	SIFT [22]	0.66	0.86	0.74
	SW	SIFT	0.71	0.80	0.76
		ME+MC	0.71	0.81	0.76
		HEVC	**0.69**	**0.83**	**0.78**
IRv2	SAE [22]	SIFT [22]	0.71	0.78	0.75
	SW	SIFT	0.69	0.82	0.75
		ME+MC	0.68	0.82	0.75
		HEVC	**0.71**	**0.83**	**0.77**
VGG16	SAE [22]	SIFT [22]	0.67	0.73	0.70
	SW	SIFT	0.69	0.83	0.75
		ME+MC	0.69	0.83	0.76
		HEVC	**0.68**	**0.85**	**0.77**

4) VSUMM DATASET – SINGLE-CNN RESULTS

Similar to Table 4, Table 6 reports the results for the VSUMM dataset. Similar to the conclusions drawn from the results in Table 4, when using the proposed HEVC feature set, our solution surpasses the CNN models when combined with stepwise regression and the HEVC feature-based frame elimination method. Additionally, the proposed stepwise regression method for dimensionality reduction achieved up to a 17% increase in performance compared to SAE across all performance metrics. Further improvement can be observed when using our proposed elimination of similar frame methods using ME+MC and HEVC features with up to 5% and 12% performance improvement, respectively, when compared to SIFT Flow across all performance metrics.

5) VSUMM DATASET – MULTI-CNN RESULTS

Table 7 shows the performance of our proposed models when using Multi-CNN features only and when using Multi-CNN features with HEVC features on the VSUMM dataset under the same run conditions as the previous runs. Again, the detection accuracy resulting from Multi-CNN surpasses the use of individual CNNs, as reported in Table 6. More noticeably, the use of Multi-CNN and HEVC resulted in an outstanding detection accuracy score, as indicated by the 0.98 F-score.

The Multi-CNN model performed better than all previous models with single CNN features by up to a 7% increase in performance across all metrics. When combining the HEVC features with the Multi-CNN features, the models showed up to a 5% score increase in performance across all metrics. Our best performing model (Multi-CNN & HEVC with

TABLE 7. Experimental results on the VSUMM dataset - Multi-CNN.

Feature Sets	Feature Reduction	Frame Elimination	Metrics		
			PPV	S	Fs
Multi-CNN	SAE [22]	SIFT [22]	0.80	0.84	0.83
	SW	SIFT [22]	0.78	0.88	0.91
		ME+MC	0.80	0.89	0.93
		HEVC	**0.82**	**0.92**	**0.96**
Multi-CNN & HEVC (proposed)	SW	SIFT [22]	0.80	0.91	0.94
		ME+MC	0.81	0.92	0.96
		HEVC	**0.85**	**0.94**	**0.98**

HEVC-based elimination) is compared with existing works in the following section.

6) VSUMM DATASET – VS. EXISTING WORKS

Figure 5 shows our best solution on the VSUMM dataset, which uses Multi-CNN and HEVC features and HEVC-based frame elimination against the state-of-the-art [22], VISCOM [44], VRHDPS [45] and VSUMM [46]. Similar to the results reported for the OVP dataset, our proposed solution for the VSUMM dataset surpasses existing work.

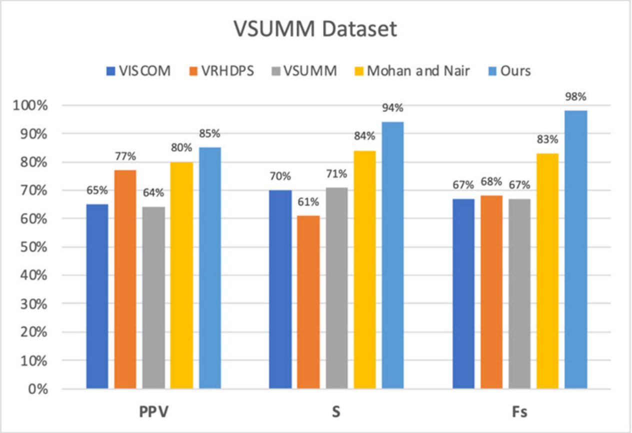


FIGURE 5. This graph shows our top performing model (Multi-CNN & HEVC and HEVC-based frame elimination) in terms of F-score on the VSUMM dataset compared with existing works in the literature.

C. DISCUSSION

Our results show an overall improvement over previous works in the literature. The advantages and limitations of the proposed solution in comparison with existing work are summarized as follows. The proposed HEVC 64 feature variables are precise and concise compared to CNN-generated features with significantly higher dimensionality. In comparison to the reviewed work, VISCOM described the video frames using novel color co-occurrence matrices [44]. VSUMM extracted the video frames attributes based on color histogram and line profiles [46]. VRHDPS used the Scale Invariant Feature Transform (SIFT) features. [22] and [39] used a novel combination of CNN features. In this work it was shown that the HEVC feature set contains rich video descriptors based on quadratic recursive splitting of coding units. These descriptors provide rich information about the spatio-temporal video content and therefore provide an excellent choice for the task at hand.

Moreover, in terms of dimensionality resolution, our solution used stepwise regression which is significantly faster than the use of auto-encoders, and yet retains enough features that are the best representative features for training and classification. The use of auto-encoders was used successfully for directionality reduction as reported in [22].

The proposed HEVC-based frame elimination avoids the high complexity of optical flow based in SIFT-descriptors which was used for frame elimination as reported in [22]. Black frames and shot boundaries were eliminated in VRHDPS [45] as they were deemed useless. In VISCOM [44], monotonic frames are eliminated based on normalized summations of squared distances between frames.

In this work, combining the proposed solutions together resulted in an average F-score of 0.93 and 0.86. Further combination of the proposed solution with multi-CNN resulted in an outstanding F-score of 0.98 for both the OVP and VSUMM datasets. However, a drawback of combining the proposed solution with multi-CNNs is that it can be computationally intensive on some computer systems. This limitation can be overcome by simply relying on the HEVC feature set alone, as it proved to identify key frames more accurately in comparison to existing work. On the other hand, it was reported in VRHDPS [45] that their solution does not require any iteration in the clustering process, rendering it an efficient algorithm. Likewise, VSUMM [46] used simple color attributes to generate quality summaries with low computational requirements.

VIII. CONCLUSION

With the surge of the Internet and surveillance footage, the need for video summarization is crucial. This research focused on the key-frame detection technique for its wide use in the literature. We proposed a feature set extracted from HEVC-coded videos. Eliminating duplicate or similar video frames was performed based on a subset of the proposed HEVC features, it was also performed based on the sum of absolute differences resulting from ME and MC. The dimensionality reduction of the feature variables was based on stepwise regression. Using the Random Forest classification, it is shown that by combining the proposed solution with Multi-CNN features, an average PPV, Sensitivity and F-score of 0.83, 0.96 and 0.98 are reported for the OVP dataset and an average of 0.85, 0.94 and 0.98 are reported for the VSUMM dataset respectively.

APPENDIX

A. ELAPSED RUN TIMES FOR THE MODELS

Setup: Intel i7 (7th gen), 16 GB RAM and GTX 1070 GPU.

TABLE 8. OVP dataset - Single-CNN, with elapsed run times added.

Feature Sets	Feature Reduction	Frame Elimination	Metrics			*Elapsed time (in seconds)*
			PPV	S	Fs	
HEVC (Proposed)	None	ME+MC	0.71	0.85	0.92	*41.3*
		HEVC	**0.87**	**0.86**	**0.93**	*41.8*
GoogleNet	SAE [22]	SIFT [22]	0.50	0.93	0.61	*803.2*
	SW	ME+MC	0.63	0.95	0.87	*46.8*
		HEVC	**0.59**	**0.96**	**0.88**	*94.0*
AlexNet	SAE [22]	SIFT [22]	0.70	0.89	0.78	*2653.3*
	SW	ME+MC	0.62	0.95	0.87	*93.9*
		HEVC	**0.59**	**0.97**	**0.88**	*262.6*
IRv2	SAE [22]	SIFT [22]	0.75	0.84	0.79	*1096.5*
	SW	ME+MC	**0.63**	**0.96**	**0.88**	*55.4*
		HEVC	0.59	0.94	0.87	*108.1*
VGG16	SAE [22]	SIFT [22]	0.68	0.61	0.64	*2615.8*
	SW	ME+MC	**0.62**	**0.74**	**0.87**	*79.5*
		HEVC	0.59	0.76	0.87	*183.3*

TABLE 9. VSUMM dataset - Single-CNN, with elapsed run times added.

Feature Sets	Feature Reduction	Frame Elimination	Metrics			*Elapsed time (in seconds)*
			PPV	S	Fs	
HEVC (Proposed)	None	ME+MC	0.71	0.82	0.75	*53.7*
		HEVC	**0.88**	**0.86**	**0.86**	*54.4*
GoogleNet	SAE [22]	SIFT [22]	0.61	0.81	0.69	*1044.2*
	SW	ME+MC	0.70	0.82	0.76	*60.8*
		HEVC	**0.68**	**0.84**	**0.78**	*122.2*
AlexNet	SAE [22]	SIFT [22]	0.66	0.86	0.74	*3449.3*
	SW	ME+MC	0.71	0.81	0.76	*122.1*
		HEVC	**0.69**	**0.83**	**0.78**	*341.3*
IRv2	SAE [22]	SIFT [22]	0.71	0.78	0.75	*1425.5*
	SW	ME+MC	0.68	0.82	0.75	*72.1*
		HEVC	**0.71**	**0.83**	**0.77**	*140.5*
VGG16	SAE [22]	SIFT [22]	0.67	0.73	0.70	*3400.5*
	SW	ME+MC	0.69	0.83	0.76	*103.4*
		HEVC	**0.68**	**0.85**	**0.77**	*238.3*

TABLE 10. Experimental results on the OVP dataset - Multi-CNN, but with elapsed run times added.

Feature Sets	Feature Reduction	Frame Elimination	Metrics			*Elapsed time (in seconds)*
			PPV	S	Fs	
Multi-CNN	SAE [22]	SIFT [22]	0.78	0.86	0.82	*6343.3*
	SW	ME+MC	0.62	0.96	0.90	*1055.6*
		HEVC	**0.59**	**0.97**	**0.93**	*1229.3*
Multi-CNN & HEVC (Proposed)	SW	ME+MC	0.65	0.95	0.97	*812.7*
		HEVC	**0.83**	**0.96**	**0.98**	*836.8*

TABLE 11. Experimental results on the VSUMM dataset - Multi-CNN, but with elapsed run times added.

Feature Sets	Feature Reduction	Frame Elimination	Metrics			*Elapsed time (in seconds)*
			PPV	S	Fs	
Multi-CNN	SAE [22]	SIFT [22]	0.80	0.84	0.83	*8246.3*
	SW	ME+MC	0.80	0.89	0.93	*1372.3*
		HEVC	**0.82**	**0.92**	**0.96**	*1598.1*
Multi-CNN & HEVC (Proposed)	SW	ME+MC	0.81	0.92	0.96	*1056.5*
		HEVC	**0.85**	**0.94**	**0.98**	*1087.9*

B. CONFUSION MATRICES

Figure 6 contains the confusion matrices for our best performing models for the OVP and VSUMM datasets. The

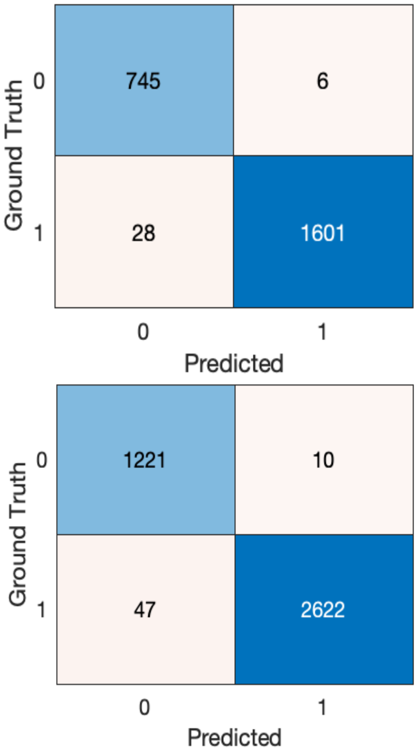


FIGURE 6. Confusion matrices of models "Multi-CNN & HEVC" with Stepwise regression feature space reduction and HEVC-based frame elimination. OVP (top), VSUMM (bottom).

best performing model for both datasets is Multi-CNN & HEVC with Stepwise regression for dimensionality reduction of the feature space and HEVC-based elimination of similar or redundant frames (i.e the models with the highest scores in tables 5 and 7). The OVP confusion matrix translates to a sensitivity score of 0.96 and the VSUMM confusion matrix translates to a sensitivity score of 0.94. Both have an F-score of 0.98, surpassing existing works in the literature.

ACKNOWLEDGMENT

This paper represents the opinions of the author(s) and does not mean to represent the position or opinions of the American University of Sharjah.

REFERENCES

[1] M. Basavarajaiah and P. Sharma, "Survey of compressed domain video summarization techniques," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–29, Nov. 2020, doi: 10.1145/3355398.

[2] T. Subba, B. Roy, and A. Pradhan, "A study on 'video summarization,'" *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 5, pp. 1–4, Jun. 2016. [Online]. Available: https://ijarcce.com/upload/2016/june-16/IJARCCE%20164.pdf, doi: 10.17148/IJARCCE.2016.56164.

[3] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: A comparative," *J. Mach. Learn. Res.*, vol. 10, nos. 66–71, p. 13, 2009.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," Sep. 2014, *arXiv:1409.4842*. Accessed: Nov. 28, 2021.

[5] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[6] V. Sze, M. Budagavi, and G. J. Sullivan, Eds., *High Efficiency Video Coding (HEVC): Algorithms and Architectures*. Cham, Switzerland: Springer, 2014, doi: 10.1007/978-3-319-06895-4.

[7] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *Computer Vision—ECCV 2016*, vol. 9911, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 766–782, doi: 10.1007/978-3-319-46478-7_47.

[8] B. Zhao, X. Li, and X. Lu, "HSA-RNN: Hierarchical structure-adaptive RNN for video summarization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7405–7414, doi: 10.1109/CVPR.2018.00773.

[9] J. Wang, W. Wang, Z. Wang, L. Wang, D. Feng, and T. Tan, "Stacked memory network for video summarization," in *Proc. 27th ACM Int. Conf. Multimedia*, Nice, France, Oct. 2019, pp. 836–844, doi: 10.1145/3343031.3350992.

[10] T. Hussain, K. Muhammad, A. Ullah, Z. Cao, S. W. Baik, and V. H. C. de Albuquerque, "Cloud-assisted multiview video summarization using CNN and bidirectional LSTM," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 77–86, Jan. 2020, doi: 10.1109/TII.2019.2929228.

[11] Z. Ji, K. Xiong, Y. Pang, and X. Li, "Video summarization with attention-based encoder–decoder networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1709–1717, Jun. 2020, doi: 10.1109/TCSVT.2019.2904996.

[12] H. Fu and H. Wang, "Self-attention binary neural tree for video summarization," *Pattern Recognit. Lett.*, vol. 143, pp. 19–26, Mar. 2021, doi: 10.1016/j.patrec.2020.12.016.

[13] B. Zhao, X. Li, and X. Lu, "TTH-RNN: Tensor-train hierarchical recurrent neural network for video summarization," *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3629–3637, Apr. 2021, doi: 10.1109/TIE.2020.2979573.

[14] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, "Video summarization using deep semantic features," in *Computer Vision—ACCV 2016*, vol. 10115, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds. Cham, Switzerland: Springer, 2017, pp. 361–377, doi: 10.1007/978-3-319-54193-8_23.

[15] Y. Jung, D. Cho, D. Kim, S. Woo, and I. S. Kweon, "Discriminative feature learning for unsupervised video summarization," Nov. 2018, *arXiv:1811.09791*. Accessed: Nov. 15, 2021.

[16] M. Rochan, L. Ye, and Y. Wang, "Video summarization using fully convolutional sequence networks," Aug. 2018, *arXiv:1805.10538*. Accessed: Nov. 15, 2021.

[17] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," Feb. 2018, *arXiv:1801.00054*. Accessed: Nov. 15, 2021.

[18] R. Agyeman, R. Muhammad, and G. S. Choi, "Soccer video summarization using deep learning," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, San Jose, CA, USA, Mar. 2019, pp. 270–273, doi: 10.1109/MIPR.2019.00055.

[19] S.-H. Zhong, J. Wu, and J. Jiang, "Video summarization via spatio-temporal deep architecture," *Neurocomputing*, vol. 332, pp. 224–235, Mar. 2019, doi: 10.1016/j.neucom.2018.12.040.

[20] T. Liu, Q. Meng, A. Vlontzos, J. Tan, D. Rueckert, and B. Kainz, "Ultrasound video summarization using deep reinforcement learning," in *Medical Image Computing and Computer Assisted Intervention—MICCAI 2020*, vol. 12263, A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz, Eds. Cham, Switzerland: Springer, 2020, pp. 483–492, doi: 10.1007/978-3-030-59716-0_46.

[21] K. Muhammad, T. Hussain, M. Tanveer, G. Sannino, and V. H. C. de Albuquerque, "Cost-effective video summarization using deep CNN with hierarchical weighted fusion for IoT surveillance networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4455–4463, May 2020, doi: 10.1109/JIOT.2019.2950469.

[22] M. S. Nair and J. Mohan, "Static video summarization using multi-CNN with sparse autoencoder and random forest classifier," *Signal, Image Video Process.*, vol. 15, no. 4, pp. 735–742, Jun. 2021, doi: 10.1007/s11760-020-01791-4.

[23] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial LSTM networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2982–2991, doi: 10.1109/CVPR.2017.318.

[24] T.-J. Fu, S.-H. Tai, and H.-T. Chen, "Attentive and adversarial learning for video summarization," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Waikoloa Village, HI, USA, Jan. 2019, pp. 1579–1587, doi: 10.1109/WACV.2019.00173.

[25] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Unsupervised video summarization via attention-driven adversarial learning," in *MultiMedia Modeling*, vol. 11961, Y. M. Ro, W.-H. Cheng, J. Kim, W.-T. Chu, P. Cui, J.-W. Choi, M.-C. Hu, and W. De Neve, Eds. Cham, Switzerland: Springer, 2020, pp. 492–504, doi: 10.1007/978-3-030-37731-1_40.

[26] K. Zhang, K. Grauman, and F. Sha, "Retrospective encoders for video summarization," in *Computer Vision—ECCV 2018*, vol. 11212, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 391–408, doi: 10.1007/978-3-030-01237-3_24.

[27] F. Wang, F. Liu, S. Zhu, L. Fu, Z. Liu, and Q. Wang, "HEVC intra frame based compressed domain video summarization," in *Proc. Int. Conf. Artif. Intell., Inf. Process. Cloud Comput. (AIIPCC)*, Sanya, China, 2019, pp. 1–7, doi: 10.1145/3371425.3371450.

[28] C. Huang and H. Wang, "A novel key-frames selection framework for comprehensive video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 577–589, Feb. 2020, doi: 10.1109/TCSVT.2019.2890899.

[29] Y. Zhao, Y. Guo, R. Sun, Z. Liu, and D. Guo, "Unsupervised video summarization via clustering validity index," *Multimedia Tools Appl.*, vol. 79, nos. 45–46, pp. 33417–33430, Dec. 2020, doi: 10.1007/s11042-019-7582-8.

[30] S. E. F. de Avila, A. P. B. Lopes, A. da Luz, and A. de Albuquerque Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognit. Lett.*, vol. 32, no. 1, pp. 56–68, Jan. 2011, doi: 10.1016/j.patrec.2010.08.004.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[32] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," Aug. 2016, *arXiv:1602.07261*. Accessed: Nov. 28, 2021.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Apr. 2015, *arXiv:1409.1556*. Accessed: Nov. 10, 2021.

[34] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012, doi: 10.1109/TCSVT.2012.2221191.

[35] M. Hassan and T. Shanableh, "Predicting split decisions of coding units in HEVC video compression using machine learning techniques," *Multimedia Tools Appl.*, vol. 78, no. 23, pp. 32735–32754, Dec. 2019, doi: 10.1007/s11042-018-6882-8.

[36] T. Shanableh, "Altering split decisions of coding units for message embedding in HEVC," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 8939–8953, Apr. 2018, doi: 10.1007/s11042-017-4787-6.

[37] S. Youssef and T. Shanableh, "Detecting double and triple compression in HEVC videos using the same bit rate," *Social Netw. Comput. Sci.*, vol. 2, no. 5, p. 406, Sep. 2021, doi: 10.1007/s42979-021-00800-8.

[38] T. Shanableh, "Saliency detection in MPEG and HEVC video using intra-frame and inter-frame distances," *Signal, Image Video Process.*, vol. 10, no. 4, pp. 703–709, Apr. 2016, doi: 10.1007/s11760-015-0798-9.

[39] J. Mohan and M. S. Nair, "Domain independent redundancy elimination based on flow vectors for static video summarization," *Heliyon*, vol. 5, no. 10, Oct. 2019, Art. no. e02699, doi: 10.1016/j.heliyon.2019.e02699.

[40] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 7th ed. Hoboken, NJ, USA: Wiley, 2018.

[41] T. Shanableh and K. Assaleh, "Feature modeling using polynomial classifiers and stepwise regression," *Neurocomputing*, vol. 73, nos. 10–12, pp. 1752–1759, Jun. 2010, doi: 10.1016/j.neucom.2009.11.045.

[42] T. Shanableh, "A regression-based framework for estimating the objective quality of HEVC coding units and video frames," *Signal Process., Image Commun.*, vol. 34, pp. 22–31, May 2015, doi: 10.1016/j.image.2015.02.008.

[43] T. Shanableh, "Detection of frame deletion for digital video forensics," *Digit. Invest.*, vol. 10, no. 4, pp. 350–360, Dec. 2013, doi: 10.1016/j.diin.2013.10.004.

[44] M. V. M. Cirne and H. Pedrini, "VISCOM: A robust video summarization approach using color co-occurrence matrices," *Multimedia Tools Appl.*, vol. 77, no. 1, pp. 857–875, Jan. 2018, doi: 10.1007/s11042-016-4300-7.

[45] J. Wu, S.-H. Zhong, J. Jiang, and Y. Yang, "A novel clustering method for static video summarization," *Multimedia Tools Appl.*, vol. 76, no. 7, pp. 9625–9641, Apr. 2017, doi: 10.1007/s11042-016-3569-x.

[46] S. E. F. de Avila, A. da Luz, Jr., A. de Albuquerque Araújo, and M. Cord, "VSUMM: An approach for automatic video summarization and quantitative evaluation," in *Proc. XXI Brazilian Symp. Comput. Graph. Image Process.*, Campo Grande, Brazil, Oct. 2008, pp. 103–110, doi: 10.1109/SIBGRAPI.2008.31.

OBADA ISSA received the B.S. degree in computer science from the American University of Sharjah, Sharjah, United Arab Emirates, in 2020, where he is currently pursuing the M.S. degree in computer engineering. His research interests include machine learning, deep learning, image processing, and video processing.

TAMER SHANABLEH (Senior Member, IEEE) was born in U.K. He received the M.Sc. degree in software engineering, in 1998, and the Ph.D. degree in electronic systems engineering from the University of Essex, in 2002.

Then, he worked as a Senior Research Officer at the University of Essex for three years, during which, he collaborated with BTexact on inventing video transcoders. Then, he joined Motorola U.K. Research Laboratories and contributed to establishing a new profile within the ISO/IEC MPEG-4 known as the Error Resilient Simple Scalable Profile. He joined the American University of Sharjah, in 2002, and is currently a Professor in computer science. During the summer breaks, he worked as a Visiting Professor at the Motorola Laboratories in five different years. He spent his sabbatical leave as a Visiting Academic at the Multimedia and Computer Vision and Laboratory, Queen Mary University of London, U.K. He studied at the University of Essex. He is currently a Professional Engineer. He has authored more than 80 publications and has six patents. His research interests include digital video processing and pattern recognition.