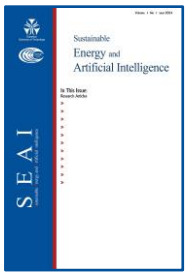
Hamedan University of Technology
Iranian Association of Electrical and Electronics Engineers

Sustainable Energy and Artificial Intelligence

Online ISSN: 3060-8015

Journal Homepage: www.enai.hut.ac.ir



Unsupervised Video Summarization Using GAN and BiLSTM-based Self-Attention Network

Alireza Gilaki[1], Roozbeh Rajabi[2*]

[1] *Faculty of Electrical and Computer Engineering, Qom University of Technology, Qom, Iran*
[2] *DITEN Department, University of Genoa, Genoa, Italy*
* *Corresponding Author: roozbeh.rajabi@unige.it*

Article Info

Article type:
Original Article

Article history:
Received 2024-11-16;
Revised 2025-01-07;
Accepted 2025-02-15.

How to cite this article:
Gilaki, A. and Rajabi, R. (2025). Unsupervised Video Summarization Using GAN and BiLSTM-based Self-Attention Network. *Sustainable Energy and Artificial Intelligence*, 1(4), 205-216.
DOI: 10.61186/seai.2411-1021

Abstract

This paper presents an approach for automated unsupervised video summarization, that means, nothing more than video is needed to train the model. The goal is to extract a sequence of frames from an input video and assign each frame a score between 0 and 1. By doing so, we can select a subset of the most informative and diverse shots to make a summarized video. We build upon the foundation of SUM-GAN, particularly SUM-GAN-SLA, which utilize Generative Adversarial Networks to compare and distinguish between the original video and its regenerated counterpart. A key contribution of our work lies in the novel biLSTM-based self-attention network that we introduce to handle the crucial scoring layer of our model. We adjusted several aspects of the model, particularly in the loss functions and learning steps, to enhance the training process and achieve superior performance compared to state-of-the-art unsupervised and even supervised methods. To ensure a fair comparison, we evaluate our proposed model using two widely used datasets: SumMe and TVSum. The experimental results highlight the effectiveness of our proposed approach in automated unsupervised video summarization, achieving a 1.2% improvement over the best-performing methods' average F-score on SumMe and TVSum datasets. Additionally, our method ranks second among state-of-the-art unsupervised methods on each dataset. Notably, the top-performing methods exhibited inconsistent results across datasets, underscoring the broader applicability of our approach to diverse types of videos. Furthermore, our method demonstrates competitive performance compared to supervised approaches, with the best supervised method surpassing our results by only 0.75%.

Keywords: Video Summarization; Generative Adversarial Networks; Attention Network; Deep Learning; BiLSTM.

1. Introduction

Today's digital age, video and multimedia usage has soared with the advent of social networks and the rapid production of visual content. Video summarization, a promising technique in the field of multimedia analysis, addresses these challenges by providing condensed representations of lengthy videos, therefore it finds applications in a wide range of fields, from video categorization and archiving to video trailer creation and content browsing. By extracting the most informative and relevant segments of a video, it enables users to quickly grasp the entire story and main ideas without having to watch the full-length video. This not only saves time but also facilitates more

efficient and effective video content consumption. It is important to distinguish video summarization from video compression, despite the presence of some similarities. While both methods aim to reduce the size of video data, video summarization focuses on content selection, capturing the essence of the video, and retaining its key elements. On the other hand, video compression primarily aims to reduce data size while preserving the whole video length with or without changing visual quality of the original video. Thus, video summarization offers a more meaningful and concise representation of the video's content, making it a valuable tool for various applications.

The output of video summarization can take two primary forms: a dynamic video summarization known as "video skim" utilized by the most articles, which includes key shots representing the most informative scenes of the original video, and a collection of the most important frames, commonly referred to as a "story-board" or static summarization utilized by some like [34], [5], etc.

The automation of video summarization tasks can be achieved through the utilization of deep networks, offering two distinct approaches: supervised and unsupervised learning. The supervised branch necessitates annotated datasets like SumMe and TVSum to train models effectively. In contrast, the unsupervised algorithms, such as the one adopted by our model, present a significant advantage by not requiring annotated data. Our focus lies in unsupervised learning, building upon the success of SUM-GAN[23], and its subsequent variations, SUM-GAN-SL[3] and SUM-GAN-AAE[1]. In [2], the author incorporates the Actor-critic method, commonly employed in various data sequence tasks such as text generation and translation, into GAN models for key fragment selection. The strength of this approach lies in its training procedure and utilization of reinforcement learning techniques.

User preferences were integrated into video summarization in the work [35]. This method leverages the user's personal image and information's semantic tree to extract user-specific images from each video. However, it's worth noting that this approach hasn't been extensively validated with popular datasets and demands a substantial amount of data. Additionally, [26] explored the use of eye tracker data to measure the motion of objects that capture a viewer's attention in a video. This model is built on human eye tracking and can identify both the extent of object motion and the salient regions that human viewers focus on. Nonetheless, it's important to acknowledge that object motion and attractiveness are not the sole determinants of a video's informativeness, making this approach not all-encompassing for video summarization. Another notable work in the realm of motion-based video summarization was introduced in [14]. This approach begins by extracting spatiotemporal features and then identifying inter-frame motion curves. It addresses shot boundaries through transition effect detection (TED) methodology, and ultimately employs an attention curve with the assistance of self-attention networks for keyframe selection.

Evaluating whether the summarized video effectively represents the full information of the original video with a good diversity, is a challenging task. Therefore, we conduct extensive training and testing on our model using SumMe and TVSum datasets, following the methodology of other unsupervised state-of-the-art models, to ensure a fair comparison. Remarkably, despite the subjective nature of video summarization by humans, where the annotator attendances can influence their selection of key moments in a video, models based on generative adversarial networks (GANs) inherently possess the ability to reconstruct the original video from its summary which is the main introduction of summarized video.

In this work, we present SUM-GAN-LSA, a novel approach that leverages the power of deep learning, generative models, attention mechanisms, and bidirectional long short-term memory (LSTM) networks to create informative and visually appealing video summaries. In comparison with previous works, our approach exhibits several notable advantages:

1) Unlike some existing methods such as [26] and [14], which primarily focus on tracking moving objects in a video, our model is not constrained to specific inter-frame or intra-frame features. Instead, it autonomously learns and selects the most salient features for more effective summarization.
2) Our method eliminates the need for additional user data or manual annotations, in contrast to [35].
3) We introduce a novel perspective by considering the informativeness of frames concerning the ability to reconstruct the main video from its summary.
4) Our summarization output, presented in the form of a concise video, is deemed more viewer-friendly compared to the conventional approach of selecting keyframes, as seen in [10].

5) The speed of training process gets better using the same hardware through our experiments on [3] and our model.

To assess the effectiveness of our method, we utilize two benchmark datasets, TVSum and SumMe, and compare our results with state-of-the-art models in the field. Through different evaluation, we demonstrate the superior performance of our approach in generating high-quality video summaries that accurately encapsulate the essence of the original videos. The combination of LSTM and self-attention mechanisms empowers our model to highlight the most salient moments and their dependencies, resulting in a comprehensive and coherent representation of the video content. Our findings underscore the potential of our SUM-GAN-LSA method as a promising solution for video summarization tasks in a wide range of applications.

2. Related Works

Leveraging the capabilities of deep learning has led to the automation of numerous tasks previously reliant on human expertise, resulting in increased efficiency in terms of both time and costs. Its applications have gained widespread popularity, encompassing areas such as disease detection, translation, and multimedia. For instance, [6] for skin cancer detection, number detection and recognition in traffic cameras like what is outlined in [31] and video summarization which is our case serve as notable examples showcasing the impactful applications of deep learning. The earliest works in video summarization [13,25,24,9] aimed to capture the most vital and diverse frames, often presented as key-frames or a storyboard. Some initial works, exemplified by [13,24,20,21,38], placed image objects as the focal point. The approach involved utilizing object detection, background removal, and eventually tracking the motion of elements. Although, works like [20,21] focused on supervised learning, generating key-frames as output, [38] introduced unsupervised learning, delving into object-level motion auto-encoding within super-segmented clips. As we know that the video summarization is a sequence-to-sequence task which serves an input of sequence of frames and the output will be a sequence, too, [37] emerged as a pioneer, employing LSTM to capture temporal dependencies across sampled frames. Building on this foundation, subsequent adjustments were made, as evidenced by [39], which resulted in video segmentation for extracting informative and user-friendly key-shots (video-skim) by using a hierarchical LSTM network and extracting the structure of video.

As video segmentation or extracting shot boundaries was a challenging task, recent works, focused on key-shot summaries, have widely embraced the successful KTS segmentation algorithm presented in [27]. This algorithm performed a temporal segmentation into semantically-consistent segments, delimited not only by shot boundaries but also general change points. Memory networks also offered valuable insights, as seen in [19], which utilized supervised learning to preserve temporal dependencies within frame sequences from 360° videos. Among numerous successful methodologies employing LSTM networks, bi-directional LSTM gained prominence, as employed in [36,23,3,16,7]. These methodologies closely mirror human summarization strategies, where the viewer initially watches the complete video and then, they find interdependencies between scenes from the beginning to the end and vice versa. Frame feature extraction stands as another complex challenge in the initial steps of video summarization. Approaches like [34] devised custom networks for feature extraction, while others utilized CNN networks. Models like [23,29] relied solely on the pool5 layer of the ImageNet-trained GoogleNet network, while some, including [3,7], introduced supplementary linear compression networks. Attention mechanisms, initially introduced for neural machine translation in [4], made their way into video summarization with significant improvements in both supervised [1] and unsupervised [7] learning. After that, the supervised attention-based architecture proposed in [16,22] further contributed to the advancement of encoding and decoding networks.

Since no more annotated datasets is needed in unsupervised learning, these kinds of methods have been more popular in the task of video summarization. They use typically generative adversarial network (GAN) [23,3,36,7] to compose a regeneration of original video and compare it with the initial video to train their weights. The techniques presented in [23,3,7] involve a scoring layer that assigns weights to frames and a variational auto-encoder (VAE) for regeneration. However, [7], which draws inspiration from [23, 3], extends this by incorporating an attention mechanism within the VAE to further enhance summarization effectiveness. It's important to note that this attention-based improvement in [7] may

not fully translate to the summarization of unseen videos, because this enhancement is solely employed during the training process and will be deactivated afterward.

Different from these variations of models, our proposed method includes a novel biLSTM-based self-attention mechanism in scoring layer taking the advantages of self-attention network and biLSTM together, outperforms many state-of-art methods even those in the supervised learning category.

3. Proposed Model

3-1. Problem Statement

Video summarization using unsupervised models with GANs revolves around the central concept of comprehending the primary video through its summary and comparing it with the regenerated version from the summary. These models process a sequence of frames extracted from various videos as input and assign importance scores (predicted probabilities) to each frame, determining its inclusion in the final summarized video. Consequently, the models select a collection of key shots, as our approach uses this method, to generate a dynamic video summary (video-skim) or a set of highly significant key frames to create a static video summary (story-board).

3-2. LSTM-based Self-Attention Network

The architecture of our model is motivated by the research introduced in [23], with specific emphasis on the SUM-GAN-SL model outlined in [3]. The pivotal element driving the summarization task is the frame selector or scoring layer, which integrates the innovative biLSTM-based self-attention (LSA) network we propose. As Fig. 1 illustrates the whole simplified diagram, our model named SUM-GAN-LSA, incorporates other networks such as VAE and discriminator during the training process. LSTM-based encoder is responsible for encoding the weighted frames to latent space and LSTM-based decoder tries to regenerate the original video from it. Finally, a 2-layer LSTM discriminator decides between the original and regenerated video that which one is real or fake.

However, once the model finds its optimal state, as depicted in Fig. 2, the entire process of summarizing new videos is executed solely by the scoring layer, with the other components no longer involved. The details of Scoring layer, as the most important part of this model, is represented in Fig. 3, where the input frame features of a video, x_i's, are 1-dimensional vectors with the size of 1024 (provided by pool5 layer of GoogleNet), then a set of frame features of a given video $X = \{\vec{x}_i\}_{i=1}^{M}$ fed into the scoring layer.

The LSA network includes a bi-directional LSTM in its first stage, enabling it to capture important events and temporal dependencies within the series of video frames in both forward and backward directions. Subsequently, we apply the self-attention mechanism, as showing in Fig. 1, to the outputs of the LSTM layer, drawing inspiration from the design proposed in VASnet [1]. In this self-attention network, three linear networks (K, Q, V) of the same size are employed to capture and compute complex, higher-order relationships between frames.

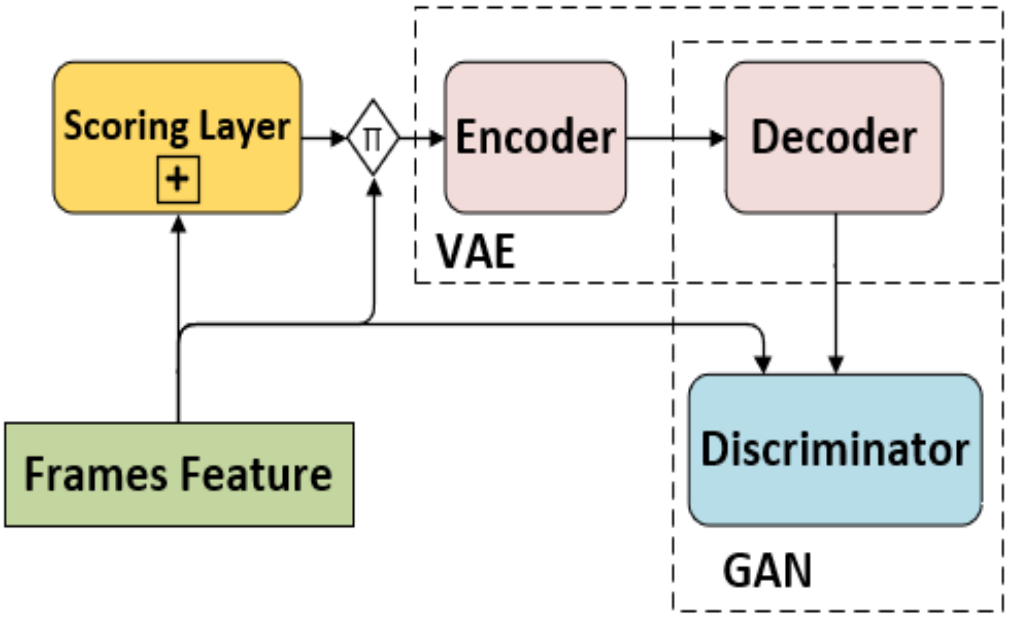


Fig. 1. A simplified diagram of the proposed model

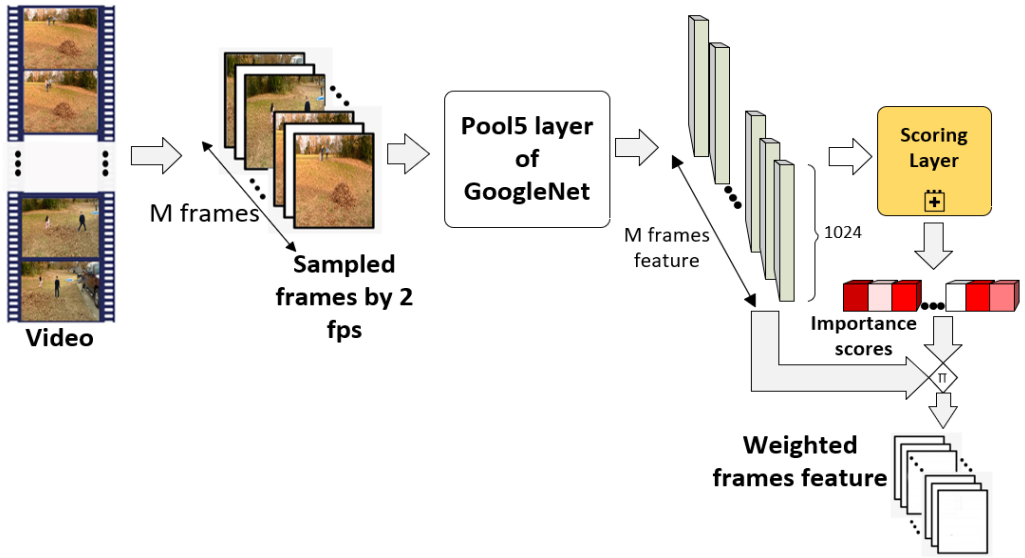


Fig. 2. Summarizer diagram, presenting the progression from video to individual frames and consequently to weighted features. The features are weighted by the importance scores assigned to each frame, obtained from the scoring layer

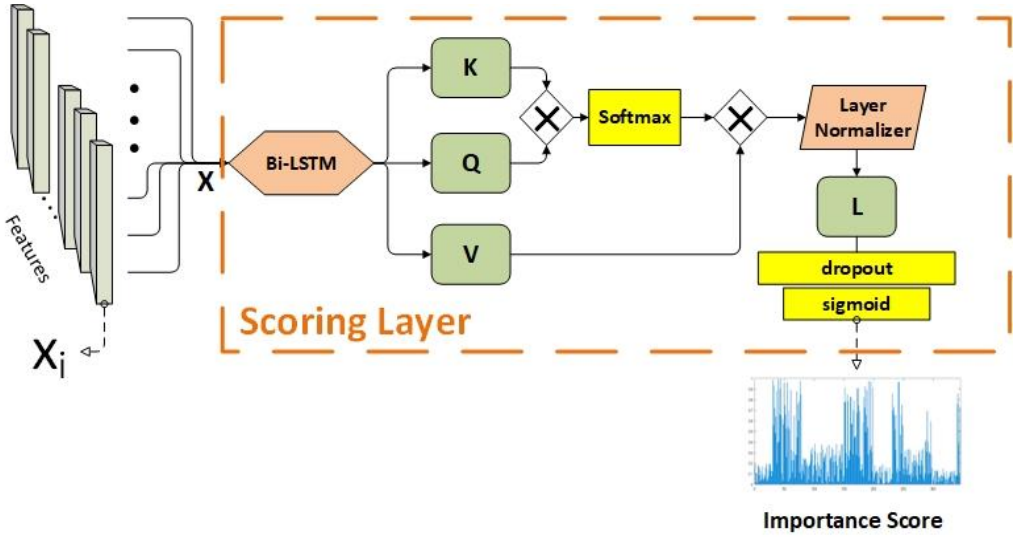


Fig. 3. Detailed diagram of our proposed scoring layer, containing bi-LSTM based self-attention network. The inputs are feature vectors and the outputs are importance scores.

The subsequent equation shows the product of the K and Q layers when X' is the result of hidden state of Bi-LSTM layer for each frame, resulting in the multiplication of their respective matrix representations:

$$Y = (KX')(QX')^T \quad (1)$$

As both K and Q share the same dimensions, Y becomes a square matrix. To introduce nonlinearity into our model, this matrix is then subjected to the Softmax function. This function is represented as below:

$$\vec{\sigma}_i = f(\vec{y}_i) = \frac{e^{y_i}}{\sum_{i=1}^{M} e^{y_i}} \quad (2)$$

The next step is to multiply the outcome with VX' and apply normalizer layer to enhance our results as follow:

$$Z = norm(\epsilon VX') \quad (3)$$

Another linear layer is utilized to map z_i to 1-by-1 score for each frame. While some modifications have been made to the self-attention network compared to VASnet, our proposed model, with or without these changes, exhibits significant efficiency and improvement, so it led us to a better model. One such unchanged parameters is the recommended multiplication of Q layer by a constant coefficient (ϵ) of 0.06, as suggested in [1]. our studies prove that any changes in this value will decrease our final F-score. Dropout layer in other approaches is considered to be 0.5 but our experiments show a good improvement when setting it to 0.1.

3-3. Loss Functions and Training Procedure

We adopt the main idea of step-wise method proposed in SUM-GAN-SL for calculating losses and train weights. A notable limitation and challenge of our method lies in the optimization of the learning objective, which involves multiple loss functions with parameters that must be carefully selected and applied. Specifically, determining the appropriate set of loss functions is critical to achieve optimal performance. This process requires extensive experimentation to evaluate the impact of various loss functions particularly the reconstruction loss 2 on the overall model performance. The inclusion or adjustment of any additional loss function necessitates thorough testing to identify the most effective integration point within the training framework. These loss functions are detailed as follows:

Sparsity Loss: Sparsity loss is designed to promote the creation of summaries with fewer frames, making them more concise and informative. By encouraging sparsity in the summarization process, the model can focus on selecting only the most relevant frames or shots, thus avoiding redundant or less informative content. For this loss, we introduce a regularization factor as a hyperparameter that influences the model's performance based on its value. The following formula shows that the mean of importance scores (S_i) of all frames, considering M-frames should get closer to regularization factor (σ).

$$\mathcal{L}_{sparsity} = |\frac{1}{M}\sum_{i=1}^{M} S_i - \sigma| \quad (4)$$

Prior loss: The prior loss can be formulated based on prior knowledge or assumptions about the desired properties of a good summary. For example, it can be designed to promote diversity, coverage, or representativeness of the selected keyframes or shots in the summary. The loss function penalizes deviations from these desired properties, encouraging the model to generate summaries that align with the predefined criteria.

The specific formulation of the prior loss can vary depending on the algorithm and the desired properties of the summary. It is often combined with other loss terms, such as reconstruction loss or adversarial loss, to jointly optimize the summarization model. The goal is to find a balance between adhering to the prior constraints and capturing the important content of the video.

Reconstruction loss 1: The decoder layer plays a crucial role in regenerating the main video from the latent space of a summarized one. To achieve this, a discriminator is employed to determine whether a given video is a regenerated version or the original video. Our model implicitly aims to perplex the discriminator layer. As highlighted in [18], the reconstruction loss is computed by measuring the Euclidean distance between the last hidden layer of the discriminator LSTM for both the original and regenerated videos. Formally, the reconstruction loss is defined as:

$$\mathcal{L}_{recon1} = |D(O) - D(R)|, \quad (5)$$

where D denotes the discriminator function, and O and R correspond to the original and regenerated videos, respectively.

Reconstruction loss 2: Despite the presence of reconstruction loss 1, which is defined in Eq. 5 and measures the Euclidean distance between the original and regenerated video after passing through the discriminator, we propose another loss function that directly computes the distance between the original and regenerated video from the generative network, also known as the decoder. However, it is important to note that this calculation can impose a heavier computational load due to the original dimension of frame features. Nevertheless, this additional loss function has shown to bring significant improvements in the results of our model.

As the following equation shows, a constant variable is multiplied with the final outcome and thus regulate the loss value. The determination of this variable can be guided by experience to achieve optimal outcomes. Striking an appropriate balance is important, where the constant 'C' must not be excessively large to overshadow other components, while simultaneously not being overly small to undermine its positive impact on the model's learning process.

$$\mathcal{L}_{recon2} = C|O - R| \quad (6)$$

Label-based Losses: Taking into account the discriminator's role in distinguishing between original and regenerated videos, its prediction probability of 0 or 1 indicates certainty whether a video is considered regenerated or original, respectively. To achieve this, we compute the difference between 0 and the probability determined by the discriminator for regenerated videos. Similarly, we calculate the loss for original videos in the same manner.

During each step of the training process, we calculate a set of losses and optimize specific networks. The selection of these two parameters at each level of optimization significantly impacts not only the final results but also the convergence and the speed of convergence. The best results were achieved when our model was trained following the forward path outlined in Table 0.

Table. 1. Three forward steps of training procedure. In each step different losses are computed and special layers are optimized.

Step	Loss Functions	Optimization
1st	$\mathcal{L}_{\text{recon2}}, \mathcal{L}_{\text{prior}}, \mathcal{L}_{\text{sparsity}}$	Scoring layer, Decoder layer, Encoder Layer
2nd	$\mathcal{L}_{\text{recon2}}, \mathcal{L}_{\text{recon1}}, \mathcal{L}_{\text{sparsity}}$	Scoring layer, Decoder layer, Encoder Layer, Discriminator
3rd	$\mathcal{L}_{\text{recon1}}, \mathcal{L}_{\text{sum\_label}}, \mathcal{L}_{\text{org\_label}}$	Scoring layer, Discriminator

4. Experiment

4-1. Datasets

Two of most popular datasets in the video summarization field are SumMe[12] and TVSum[32]. SumMe dataset containing 25 MP4 videos of diverse topics, including sports events, talks and holidays. Their length also is between 1.5 to 6.5 minutes with about 15 human-created annotation in the form of key-fragment scores for each. On the other hand, TVSum, also known as Title-based Video Summarization, is another widely used dataset for video summarization research. It consists of 50 videos from various genres such as news, how-to, documentary, vlog, and egocentric videos, etc. The dataset also provides about 1,000 annotations of shot-level importance scores, with 20 annotations per video, obtained through crowdsourcing. The videos in the TVSum dataset have a duration ranging from 2 to 10 minutes. This duration range reflects the typical length of videos encountered in television broadcasts and online video platforms. Both datasets provide diversity in content genres, make it valuable and a good evaluation platform for different video summarization techniques.

4-2. Evaluation Metrics

For a fair comparison with other state-of-the-art techniques in video summarization, we adopt the proposed evaluation method in [37] corresponding to key-shot importance score to check the similarity between our generated summary and ground truth. For a given video, let M and G be the generated summary by Model and ground truth summary, respectively. The precision (P) and recall (R) parameters are commonly used to measure the overlap between the M and G groups of frames, which are divided by the duration of the M summary for P and the duration of the G summary for R. The F-score can then be calculated as follows:

$$F = \frac{2PR}{P+R} * 100 \quad for \quad P = \frac{M \cap G}{M} \quad and \quad R = \frac{M \cap G}{G} \tag{7}$$

Another explanation of F-score is in the form of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN):

True Positive (TP): These are frames correctly identified as part of the summary. The model accurately recognizes them as relevant.

True Negative (TN): These are frames correctly classified as not part of the summary. The model correctly identifies them as non-relevant.

False Positive (FP): These are frames incorrectly classified as part of the summary. They are false alarms, as they are not actually relevant.

False Negative (FN): These are frames that were not selected as part of the summary but should have been. The model fails to identify these frames, resulting in the loss of important information.

Consequently, we can re-write this F-score formulation:

$$F = \frac{TP}{TP + \frac{1}{2}(FN + FP)} * 100 \tag{8}$$

In order to compare our generated summaries with the dataset annotations, we have two approaches: 1) comparing our outputs with a single choice of user annotation, or 2) comparing our outputs with all annotations available (approximately 15 user summaries for SumMe and 20 for TVSum) and then averaging the computed F-scores in the case of TVSum, or selecting the maximum F-score in the case of SumMe, as recommended by [4]. Given the lack of a definitive answer for the video summarization task and the potential influence of personal preferences on annotators' summaries, many studies adopt the second approach, wherein the model is compared with the opinions of different annotators. In contrast, the first approach used by [27] poses a challenge of selecting a single sample of user summaries, where our outcomes may vary across different users. Although many articles have adopted the first method, but based on these reasons, we choose the second one.

As previously mentioned, there is a difference in the level of importance scores between TVSum, which operates at the frame-level, and SumMe, which operates at the shot-level. In order to facilitate a fair comparison between the two datasets, we employ a conversion method to align them at the shot-level. This conversion process is based on the approach presented in [27], where we utilize the KTS algorithm [23] to segment the video into sequences of shots and extract shot

boundaries. Subsequently, we calculate the average importance score of all frames within each shot to derive the shot-level importance score. Next, taking the knapsack algorithm into account, we can identify key shots to make the summary with the length of 15% of original video duration. By applying this method, we ensure a consistent level of comparison between TVSum and SumMe, allowing for meaningful analysis and evaluation.

4-3. Training and Implementation Details

Before processing video data, it is necessary to convert it into a sequence of frames. However, extracting all frames from a video can be computationally expensive. To address this, we employ either uniform or sparse sampling techniques. Sparse sampling involves selecting frames based on specific criteria such as frame transitions or I-frames, while uniform sampling involves selecting frames at a constant rate. In our case, we follow the recommendation in [27] and sample videos at 2 frames per second (fps). This approach strikes a balance between capturing essential frames and reducing computational load, while also ensuring that important frames are not lost, even in videos with minimal movement and frame changes. To extract features and reduce the size of frames, we make use of output of pool5 layer of GoogleNet [33] network which has been pre-trained on ImageNet [30] with the output size of 1024 for each frame.

In previous studies [23, 3], the choice of a learning rate of 10^{-4} and a hidden size of 500 has been reported. However, in our experiments, we observed that our sub-networks achieved the best performance when using a learning rate of 0.001 and a hidden size of 256. Prior to training the model, both datasets were divided into five random splits. Each split allocated 20% of the data for testing and 80% for training. This process was repeated five times over 100 epochs, allowing the model to be trained and tested on different subsets of the data. The average of results obtained from these iterations was then reported as the final performance of the model.

5. Results and Discussion

In this section, we aim to identify the optimal hyperparameter values that lead to the highest and best performance of our model. We then proceed to compare our results with the state-of-the-art models in both supervised and unsupervised learning approaches.

Similar to the evaluation process in [3], we conducted an assessment of SUM-GAN-LSA (our model) to determine the most suitable regularization factor for the sparsity loss calculation. The model was tested on both the SumMe and TVSum datasets using various regularization factor values. The other values out of the presented range significantly reduced the results. Table 1 reveals that the peak F-score value was achieved when the regularization factor σ was set to 0.1 for SumMe and 0.5 for TVSum. Nevertheless, choosing a value of 0.1 provides an acceptable performance in both datasets.

Table 2. The impact of regularization factor values on final F-scores in both datasets. The optimal results were achieved with values of 0.1 for SumMe and 0.5 for TVSum.

Regularization Factor (σ)	SumMe	TVSum
0.05	48.8	59.3
0.1	**49.0**	60.6
0.2	48.9	60.1
0.4	47.9	60.1
0.5	47.8	**60.7**
0.6	48.0	59.8

As previously mentioned, the proposed reconstruction loss 2 has a significant impact on both the training process and the overall performance of our model. Since reconstruction loss 2 carries more weight in the total loss, we introduced a constant coefficient to normalize it appropriately. To determine the optimal coefficient value, we conducted extensive evaluations on two datasets, as depicted in Fig. 4. The results clearly indicate that the best performance is achieved when the coefficient is set to 0.08 for both datasets. Notably, omitting this loss function (setting the coefficient to zero) leads to a significant reduction in performance, resulting in a decrease of 2% in SumMe and 2.6% in TVSum.

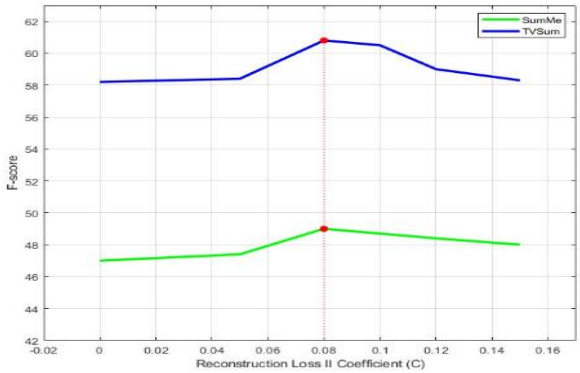


Fig. 4. The effect of changing the reconstruction loss II coefficient in the F-score in SumMe and TVSum. the best outcomes occur when it is set to 0.08

In our pursuit of optimizing the model's performance, we explored various settings that proved to be beneficial. While SUM-GAN-SL recommended neural network hidden layers of 500, we made the decision to use a hidden layer of size 256, which resulted in improved performance. Additionally, the introduction of a compression layer in the first stage of SUM-GAN-SL had initially shown to increase F-scores. However, in our model, it had the opposite effect, leading to a decrease in performance. The main reason behind this discrepancy can be attributed to the inherent LSTM-based self-attention network, which already possesses adequate linear layers. By adding an extra linear layer, namely the compression layer, we add more weights to the model. Considering these parameters and conditions, we compare SUM-GAN-LSA with the best-performing unsupervised and supervised methods in Tab. 3 and Tab. 4, respectively.

Table. 3. Comparing the final F-score of SUM-GAN-LSA with the best performing unsupervised models. It shows that our proposed method out-performs all of represented models.

Models	SumMe	TVSum	Average
Online-motion-AE [38]	37.7	51.5	44.6
unpairedVSN $_{unsup}$[28]	47.5	55.6	51.55
SUMFCN $_{sup}$ [29]	41.5	51.7	46.6
SUM-GAN $_{unsup}$[23]	38.7	50.8	44.75
SUM-GAN-SL [3]	47.3	58.0	52.65
SUM-GAN-AAE [7]	48.9	58.3	53.6
Tesselation $_{unsup}$[17]	41.4	**64.1**	52.75
SUM-GAN-LSA(ours)	**49.0**	60.6	**54.8**

As depicted in Tab. 3, our approach consistently outperforms all prior unsupervised methods on the SumMe dataset. Notably, Tessellation[17] achieved a remarkable result of 64.1 in the TVSum dataset; however, its efficiency was not good when applied to the SumMe dataset. To ensure equitable comparison, we introduced an additional metric that computes the average performance across both the SumMe and TVSum datasets. Remarkably, SUM-GAN-LSA emerged as the most proficient model under this comprehensive assessment.

Our model's performance is notable even when compared to supervised methods, which inherently leverage significantly more annotated data. Our approach demonstrates remarkable results, closely approximating the performance of state-of-the-art models, as we see in Tab. 4. The VASNet[1] method stands out as a superior solution, with a little marginal difference of 0.7 in the SumMe dataset and 0.8 in the TVSum dataset, in comparison to our model. It is important to emphasize that, after the model is trained, the summarization layer remains relatively computationally intensive. Due to the structure of our model, the video summarization process requires slightly more time compared to other methods.

Table 4. Comparison of F-scores between SUM-GAN-LSA and state-of-the-art (SoA) supervised models. Our method ranks second among SoA models, with bold values denoting optimal performance across all models and plus sign results indicating superior performance compared to ours.

Models	SumMe	TVSum	Average
unpairedVSN $_{sup}$ [28]	48.0	56.1	44.6
SUMFCN $_{sup}$ [29]	47.5	56.8	52.15
SUM-GAN $_{sup}$ [23]	38.7	50.8	44.75
Multi-Stage [15]	46.1	60.0	53.05
MAVS [8]	40.3	**66.8**(+)	53.55
Tesselation $_{sup}$ [17]	37.2	63.4(+)	50.3
vsLSTM [11]	37.6	54.2	45.9
VASNet [1]	**49.7**(+)	61.4(+)	**55.55**(+)
SUM-GAN-LSA (ours)	49.0	60.6	54.8

6. Conclusion

In this study, we have introduced an innovative approach to video summarization, building upon the foundation of the SUM-GAN-SL method and incorporating a novel biLSTM-based self-attention network. The integration of this network into the scoring layer of our model synergizes the strengths of both LSTM and attention networks, enabling our model to effectively summarize unseen videos. Our introduced loss functions and modifications in the loss calculation at various stages have significantly contributed to the remarkable results achieved. We not only evaluated our final F-scores against recent models using SumMe and TVSum datasets individually but also calculated the average of two datasets results. This average serves as a comprehensive measure of overall model performance, providing a comprehensive assessment of model effectiveness. At the end, our findings clearly demonstrate the superior performance of SUM-GAN-LSA when compared to the state-of-the-art video summarization techniques, spanning both unsupervised and most supervised approaches. In anticipation of future developments, we recommend incorporating a reinforcement learning model into the training process. Additionally, enhancing the preprocessing step by selectively sampling most diverse frames in a non-uniform manner can improve the model's

ability to select the frames with minimal correlation. Furthermore, training the proposed network on distinct categories individually is expected to yield improved results.

Declarations

- Conflict of interest: The authors declare that they have no conflict of interest.
- Availability of data and materials: The datasets used for the current study are TVSum available at https://github.com/yalesong/tvsumhttps://github.com/yalesong/tvsum, and SumMe available at https://cove.thecvf.com/datasets/615https://cove.thecvf.com/datasets/615

References

[1] Tonge, A., & Thepade, S. D. (2022). Creating Video Visual Storyboard with Static Video Summarization using Fractional Energy of Orthogonal Transforms. *International Journal of Advanced Computer Science and Applications*, *13*(9).

[2] Chen, S. N. (2017). Storyboard-based accurate automatic summary video editing system. *Multimedia Tools and Applications*, *76*(18), 18409-18423.

[3] Mahasseni, B., Lam, M., & Todorovic, S. (2017). Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 202-211).

[4] Apostolidis, E., Metsai, A. I., Adamantidou, E., Mezaris, V., & Patras, I. (2019, October). A stepwise, label-based approach for improving the adversarial training in unsupervised video summarization. In *Proceedings of the 1st International Workshop on AI for Smart TV Content Production, Access and Delivery* (pp. 17-25).

[5] Apostolidis, E., Adamantidou, E., Metsai, A. I., Mezaris, V., & Patras, I. (2019, December). Unsupervised video summarization via attention-driven adversarial learning. In *International Conference on multimedia modeling* (pp. 492-504). Cham: Springer International Publishing.

[6] Apostolidis, E., Adamantidou, E., Metsai, A. I., Mezaris, V., & Patras, I. (2020). AC-SUM-GAN: Connecting actor-critic and generative adversarial networks for unsupervised video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, *31*(8), 3278-3292.

[7] Yin, Y., Thapliya, R., & Zimmermann, R. (2016). Encoded semantic tree for automatic user profiling applied to personalized video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(1), 181-192.

[8] Paul, M., & Salehin, M. M. (2018). Spatial and motion saliency prediction method using eye tracker data for video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(6), 1856-1867.

[9] Huang, C., & Wang, H. (2019). A novel key-frames selection framework for comprehensive video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(2), 577-589.

[10] Guan, G., Wang, Z., Lu, S., Da Deng, J., & Feng, D. D. (2012). Keypoint-based keyframe selection. *IEEE Transactions on circuits and systems for video technology*, *23*(4), 729-734.

[11] Faghihi, A., Fathollahi, M., & Rajabi, R. (2024). Diagnosis of skin cancer using VGG16 and VGG19 based transfer learning models. *Multimedia Tools and Applications*, *83*(19), 57495-57510.

[12] Shahidi Zandi, M., & Rajabi, R. (2022). Deep learning based framework for Iranian license plate detection and recognition. *Multimedia Tools and Applications*, *81*(11), 15841-15858.

[13] Hong, R., Tang, J., Tan, H. K., Yan, S., Ngo, C., & Chua, T. S. (2009, October). Event driven summarization for web videos. In *Proceedings of the first SIGMM workshop on Social media* (pp. 43-48).

[14] Ngo, C. W., Ma, Y. F., & Zhang, H. J. (2005). Video summarization and scene detection by graph modeling. *IEEE Transactions on circuits and systems for video technology*, *15*(2), 296-305.

[15] Ma, Y. F., Lu, L., Zhang, H. J., & Li, M. (2002, December). A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on Multimedia* (pp. 533-542).

[16] Gong, B., Chao, W. L., Grauman, K., & Sha, F. (2014). Diverse sequential subset selection for supervised video summarization. *Advances in neural information processing systems*, *27*.

[17] Lee, Y. J., Ghosh, J., & Grauman, K. (2012, June). Discovering important people and objects for egocentric video summarization. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 1346-1353). IEEE.

[18] Liu, D., Hua, G., & Chen, T. (2010). A hierarchical visual model for video object summarization. *IEEE transactions on pattern analysis and machine intelligence*, *32*(12), 2178-2190.

[19] Zhang, Y., Liang, X., Zhang, D., Tan, M., & Xing, E. P. (2020). Unsupervised object-level video summarization with online motion auto-encoder. *Pattern Recognition Letters*, *130*, 376-385.

[20] Zhang, K., Chao, W. L., Sha, F., & Grauman, K. (2016, September). Video summarization with long short-term memory. In *European conference on computer vision* (pp. 766-782). Cham: Springer International Publishing.

[21] Zhao, B., Li, X., & Lu, X. (2018). Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7405-7414).

[22] Potapov, D., Douze, M., Harchaoui, Z., & Schmid, C. (2014, September). Category-specific video

summarization. In *European conference on computer vision* (pp. 540-555). Cham: Springer International Publishing.

[23] Lee, S., Sung, J., Yu, Y., & Kim, G. (2018). A memory network approach for story-based temporal summarization of 360 videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1410-1419).

[24] Yuan, L., Tay, F. E., Li, P., Zhou, L., & Feng, J. (2019, July). Cycle-SUM: Cycle-consistent adversarial LSTM networks for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, No. 01, pp. 9143-9150).

[25] Ji, Z., Xiong, K., Pang, Y., & Li, X. (2019). Video summarization with attention-based encoder–decoder networks. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(6), 1709-1717.

[26] Fajtl, J., Sokeh, H. S., Argyriou, V., Monekosso, D., & Remagnino, P. (2018, December). Summarizing videos with attention. In *Asian conference on computer vision* (pp. 39-54). Cham: Springer International Publishing.

[27] Rochan, M., Ye, L., & Wang, Y. (2018). Video summarization using fully convolutional sequence networks. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 347-363).

[28] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[29] Liu, Y. T., Li, Y. J., Yang, F. E., Chen, S. F., & Wang, Y. C. F. (2019, September). Learning hierarchical self-attention for video summarization. In *2019 IEEE international conference on image processing (ICIP)* (pp. 3377-3381). IEEE.

[30] Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2016, June). Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning* (pp. 1558-1566). PMLR.

[31] Gygli, M., Grabner, H., Riemenschneider, H., & Van Gool, L. (2014, September). Creating summaries from user videos. In *European conference on computer vision* (pp. 505-520). Cham: Springer International Publishing.

[32] Song, Y., Vallmitjana, J., Stent, A., & Jaimes, A. (2015). Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5179-5187).

[33] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[34] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*(3), 211-252.

[35] Rochan, M., & Wang, Y. (2019). Video summarization by learning from unpaired data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7902-7911).

[36] Kaufman, D., Levi, G., Hassner, T., & Wolf, L. (2017). Temporal tessellation: A unified approach for video analysis. In *Proceedings of the IEEE international conference on computer vision* (pp. 94-104).

[37] Huang, S., Li, X., Zhang, Z., Wu, F., & Han, J. (2018). User-ranking video summarization with multi-stage spatio–temporal representation. *IEEE Transactions on Image Processing*, *28*(6), 2654-2664.

[38] Feng, L., Li, Z., Kuang, Z., & Zhang, W. (2018, October). Extractive video summarizer with memory augmented neural networks. In *Proceedings of the 26th ACM international conference on Multimedia* (pp. 976-983).

[39] Gygli, M., Grabner, H., & Van Gool, L. (2015). Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3090-3098).

Biography

Alireza Gilaki received the M.Sc. degree from Qom University of Technology, Qom, Iran, in 2023, in Electrical engineering majoring in telecommunications. His current research interests include deep learning, video processing, and digital signal and image processing.

Roozbeh Rajabi received his B.Sc. degree in Electrical Engineering from the Iran University of Science and Technology, Tehran, Iran, in 2007, and his M.Sc. degree in Biomedical Engineering (Bioelectric) from Tarbiat Modares University, Tehran, Iran, in 2009. He completed his Ph.D. in Communication Systems Engineering at Tarbiat Modares University in 2014. He is currently an Assistant Professor at the University of Genoa (UniGe), Genoa, Italy. His research interests include hyperspectral data analysis, pattern recognition, and digital signal and image processing. He is a Senior Member of IEEE.