Unsupervised Video Summarization via Attention-Driven Adversarial Learning

Evlampios Apostolidis[1,2], Eleni Adamantidou[1], Alexandros I. Metsai[1], Vasileios Mezaris[1], and Ioannis Patras[2]

[1] Centre for Research and Technology Hellas, Thermi-Thessaloniki, Greece
{apostolid, adamelen, alexmetsai, bmezaris}@iti.gr
[2] School of EECS, Queen Mary University of London, London, UK
i.patras@qmul.ac.uk

Abstract. This paper presents a new video summarization approach that integrates an attention mechanism to identify the significant parts of the video, and is trained unsupervisingly via generative adversarial learning. Starting from the SUM-GAN model, we first develop an improved version of it (called SUM-GAN-sl) that has a significantly reduced number of learned parameters, performs incremental training of the model's components, and applies a stepwise label-based strategy for updating the adversarial part. Subsequently, we introduce an attention mechanism to SUM-GAN-sl in two ways: i) by integrating an attention layer within the variational auto-encoder (VAE) of the architecture (SUM-GAN-VAAE), and ii) by replacing the VAE with a deterministic attention auto-encoder (SUM-GAN-AAE). Experimental evaluation on two datasets (SumMe and TVSum) documents the contribution of the attention auto-encoder to faster and more stable training of the model, resulting in a significant performance improvement with respect to the original model and demonstrating the competitiveness of the proposed SUM-GAN-AAE against the state of the art.[1]

Keywords: Video summarization · Unsupervised learning · Attention mechanism · Adversarial learning.

1 Introduction

Recent advances in video capturing and storage technology, combined with the widespread use of social networks (e.g. Facebook) and video hosting platforms (e.g. YouTube), facilitate the recording and sharing of huge volumes of video content. Thousands of hours of video are uploaded every day on the Web, aiming to attract the viewers' attention. However, in several cases, browsing through long videos to find the content that a viewer prefers is a highly time-consuming and tedious process. Hence, the provision of a concise summary that conveys the main concept of the video enables the viewer to quickly grasp an idea without having to watch the entire content, thus allowing time-efficient browsing of large video collections and increasing the potential of a video to be consumed.

[1] Software publicly available at: https://github.com/e-apostolidis/SUM-GAN-AAE

Several methods have been proposed to automate video summarization, and the researchers' focus was recently attracted by deep learning architectures. In this direction, annotated datasets were built to facilitate training and evaluation. However, since video summarization is a highly-subjective task we argue that supervised learning approaches, which rely on the use of a single ground-truth summary, cannot fully explore the learning potential of such architectures. Hence, we focused on developing an unsupervised method for video summarization. Starting from [16] and building on a variation of this model [5], we scrutinized features of the architecture and the training process that could be fine-tuned to improve the model's performance. The resulting architecture SUM-GAN-sl (Section 3.1) has a reduced number of parameters, updates the model's components in an incremental manner, and follows a stepwise label-based approach for training the adversarial part. Then, inspired by the efficiency of attention mechanisms in spotting the attractive parts of a data sequence, we extended SUM-GAN-sl by: a) directly introducing an attention layer within the variational auto-encoder of the model (SUM-GAN-VAAE; Section 3.2), and b) replacing this component with a deterministic attention auto-encoder (SUM-GAN-AAE; Section 3.3). Experiments on the SumMe and TVSum datasets (Section 4), document the improvements that are attained in comparison to the original SUM-GAN model, show the inability of variational attention to enhance the training capacity of the architecture, highlight the contribution of the attention auto-encoder to faster and more stable training of the model, and show the competitiveness of the proposed SUM-GAN-AAE architecture against the state of the art.

2 Related Work

Various approaches to video summarization were introduced over the last couple of decades, with the majority being trained supervisingly using ground-truth data. For the sake of space, we report here only on methods exploiting the learning efficiency of neural networks, which represent the current state of the art; while, special focus is put on approaches utilizing attention mechanisms. A number of supervised algorithms (e.g. [17]) use Convolutional Neural Networks (CNN) to extract information about the video semantics and use it to learn to identify the most suitable parts of the video. [21] tackles video summarization as a sequence labeling problem and performs keyframe selection using fully convolutional sequence models. [7] combines a soft, self-attention network with a 2-layer fully connected network to process the CNN features of the video frames and compute frame-level importance scores that are used for key-fragment selection.

Other supervised techniques utilize Recurrent Neural Networks (RNN) (e.g. Long Short-Term Memory (LSTM) units [12]) to capture temporal dependencies over sequential data. This idea was first introduced in [27], and further expanded in [30] and [31], with hierarchies of LSTMs that extract and encode data about the video structure, and identify the key-fragments of the video. [28] combines LSTMs with Dilated Temporal Relational units to capture long-range dependencies at different temporal windows, while training relies on the generative adversarial framework. Other approaches introduce attention mechanisms

to identify the most attractive parts of the video. [13] formulates video summarization as a sequence-to-sequence learning problem and proposes an LSTM-based encoder-decoder network with an intermediate attention layer. In [9], the typical encoder-decoder seq2seq model is replaced by a special attention-based seq2seq model that defines and ranks the different fragments of the video, and is combined with a 3D-CNN classifier which judges whether a fragment is from a ground-truth or a generated summary. [8] introduces an architecture with memory augmented networks for global attention modeling, and tackles video summarization by estimating the temporal dependency across the entire video.

Contrary to the above supervised approaches, a few unsupervised methods were proposed too. [16] selects a sparse subset of representative keyframes by training a summarizer to minimize the distance between videos and a distribution of their summaries using a Generative Adversarial Network (GAN). Similarly, [25] aims to maximize the mutual information between summary and video using an information-preserving metric, two trainable discriminators and a cycle-consistent adversarial learning objective. [32] formulates video summarization as a sequential decision-making process and trains a deep summarization network to produce diverse and representative video summaries via reinforcement learning. [29] extracts key motions of appearing objects and learns to produce an object-level video summarization in an unsupervised manner. Finally, [20] proposes an adversarial process that learns a mapping function from raw videos to human-like summaries, based on professional summary videos available online.

The contributions of our work are: i) the introduction of an attention mechanism in an unsupervised learning framework, whereas all previous attention-based summarization methods ([7–9, 13]) were supervised; ii) the investigation of integrating attention into a variational auto-encoder for video summarization purposes; and iii) the use of attention to guide the generative adversarial training of the model, rather than using it to rank the video fragments as in [9].

3 Proposed Approach

3.1 Building on adversarial learning

The starting point of our work is the unsupervised method of [16]. This algorithm selects the video keyframes by minimizing the distance between the deep feature representations of the original video and a reconstructed version of it based on the selected keyframes. The difficulty in defining a suitable similarity threshold was tackled by using adversarial learning and introducing a trainable discriminator network. So, the goal was to train the summarizer (that contains the generator) in order to maximally confuse the discriminator when trying to distinguish the original from the reconstructed video; a condition that indicates a highly representative keyframe summary. Based on an implemented variation of this model [5] (used to evaluate SUM-GAN when summarizing 360° videos [15]), we scrutinized features of the architecture and the training process that could be fine-tuned to improve the model's performance. As depicted in the block-diagram of Fig. 1, we developed a new model (called SUM-GAN-sl and presented in [1])

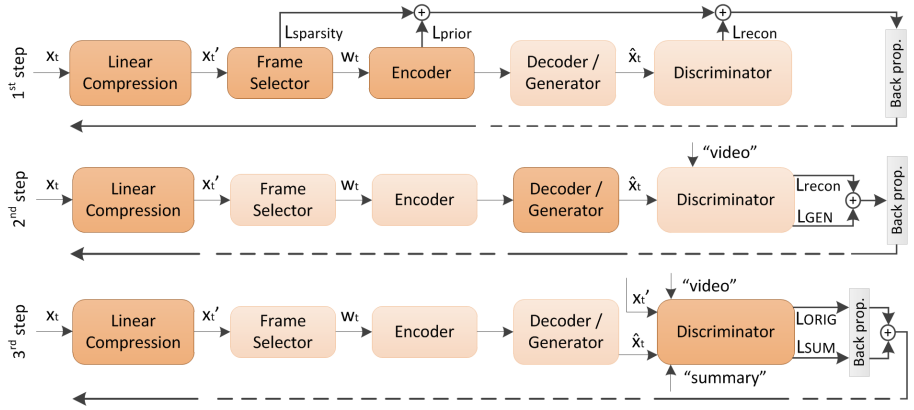

Fig. 1. Incremental training of SUM-GAN-sl. Adversarial learning follows a stepwise label-based approach. Dark-coloured boxes show updated parts in each backward pass.

that: i) contains a linear compression layer which reduces the size of the input feature vectors and the number of learned parameters, ii) follows an incremental approach for training the model's components, and iii) applies a stepwise label-based learning strategy for the adversarial part of the architecture.

Given a video of T frames, x_t, x'_t, and $\hat{x}_t$ represent the original, compressed and reconstructed feature vectors respectively, with $t \in [1, T]$. In the first step of the training process the algorithm performs a forward pass of the model, computes the L_{recon}, L_{prior} and $L_{sparsity}$ losses, and updates the frame selector, the encoder and the linear compression layer. In the second step it performs a forward pass of the partially updated model, computes the L_{recon} and L_{GEN} losses, and updates the decoder and the linear compression layer. The third step is implemented in a fine-grained, stepwise manner based on a strategy used in [19] for unsupervised representation learning that targets the task of image generation. In particular, the compressed feature vectors of the original video ($x'_t,\ t \in [1, T]$) pass through the discriminator, the L_{ORIG} loss is calculated using a label-based approach and the gradients for this loss are computed via a backward pass. Subsequently, the same feature vectors pass through the updated summarizer, the reconstructed features ($\hat{x}_t,\ t \in [1, T]$) are forwarded to the discriminator, the L_{SUM} loss is calculated and the gradients from both the original video and the summary-based reconstructed version of it are accumulated with another backward pass. With the gradients accumulated, the algorithm updates the discriminator and the linear compression layer. This fine-grained computation of gradients helps the discriminator to develop higher discrimination efficiency.

With respect to the utilized losses for training the model, L_{recon}, L_{prior} and $L_{sparsity}$ are computed as in [16]. However, instead of using the L_{GAN} loss of the SUM-GAN model, we adopt a label-based approach where label "1" denotes the original video and label "0" the video summary. Given these labels, the generator is trained based on the following loss:

$$L_{GEN} = (1 - p(\hat{\mathbf{x}}))^2 \tag{1}$$

With $p(\hat{\mathbf{x}})$ being the soft-max output of the discriminator for the reconstructed video ($\hat{\mathbf{x}} = \{\hat{x}_t\}_{t=1}^{T}$), L_{GEN} is used to minimize the Mean Squared Error (MSE) between the original video label and the assigned probability to the reconstructed video. Similarly, the discriminator is trained based on the following losses:

$$L_{ORIG} = (1 - p(\mathbf{x'}))^2 \text{ and } L_{SUM} = (p(\hat{\mathbf{x}}))^2 \tag{2}$$

With $p(\mathbf{x'})$ being the soft-max output of the discriminator for the original video ($\mathbf{x'} = \{x'_t\}_{t=1}^{T}$) and $p(\hat{\mathbf{x}})$ as described above, L_{ORIG} is used to minimize the MSE between the original video label and the assigned probability to the original video, while L_{SUM} is used to minimize the MSE between the summary label and the assigned probability to the summary-based reconstruction of the video. This stepwise, label-based learning approach allows better training of the adversarial part of the model, via a more fine-grained update of the discriminator's gradients and the use of a more strictly defined learning task for the generator.

Given a trained model, the key-fragment summary for an unseen video is created based on the following process. The CNN feature vectors of the video frames pass through the linear compression layer and the frame selector, which computes frame-level importance scores. Then, after having the video segmented using the KTS algorithm [18] (other methods (e.g. [2, 3]) could be used too), fragment-level importance scores are calculated by averaging the scores of each fragment's frames. Finally, following the approach of several summarization algorithms (e.g. [13, 22, 27, 32]), the summary is created by selecting the fragments that maximize the total importance score, provided that the summary does not exceed 15% of video duration, by solving the 0/1 Knapsack problem.

3.2 Introducing an attention mechanism

The idea behind the use of an attention mechanism for video summarization is to implement a gradual decision-making approach that bases the selection of a piece of data from a data sequence on the previously seen ones. Inspired by [13], we extended the SUM-GAN-sl model (Section 3.1) by integrating an attention layer within the variational auto-encoder of the architecture. A recent work (see [4]) that aimed to build a method for natural language modeling, examined different settings for this integration and described the bypassing effect that the traditional (deterministic) attention mechanism has on the VAE's functionality, since the latter has no impact in the process. To avoid this effect, the authors of [4] proposed a variational attention mechanism where the attention vector is also modeled as Gaussian distributed random variables. Hence, the original VAE is extended as shown in Fig. 2, in order to compute a latent variable also for the attention vector and use it when generating the decoded representation of the input sequence. Based on the above, we extended the SUM-GAN-sl model with variational attention, forming the SUM-GAN-VAAE architecture. In particular, the attention weights of each frame were considered as random variables and a latent space was computed by the VAE for these values, too. Finally, the decoding part of this component was modified in order to update its hidden states based on both latent spaces (computed for the encoder's output and the attention values) during the reconstruction of the video.

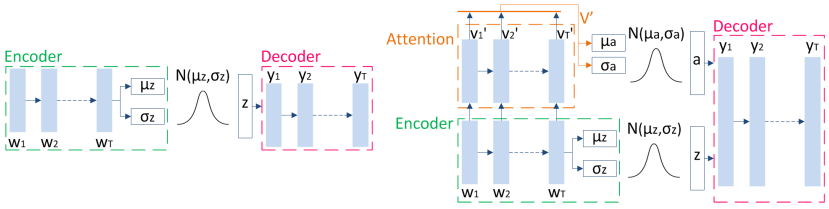

Fig. 2. Going from variational (left) to variational attention auto-encoder (right).

3.3 Introducing an attention auto-encoder

The second alternative examined for integrating an attention mechanism to the SUM-GAN-sl model is based on the supervised attention-based encoder-decoder network of [13]. Since deterministic attention bypasses the functionality of VAE, the latter is entirely replaced by an attention auto-encoder (AAE) network. The architecture of the new model (called SUM-GAN-AAE) is shown in Fig. 3. Given the t_{th} frame of a video of T frames and using the notations from Section 3.1, $s_t,\ t \in [1,T]$ refers to the computed importance score by the frame selector, $w_t,\ t \in [1,T]$ corresponds to its weighted feature vector ($s_t \otimes x'_t$, where $\otimes$ denotes element-wise matrix multiplication) that is passed to the attention auto-encoder, and $\hat{x}_t,\ t \in [1,T]$ represents the reconstructed feature vector.

Focusing on the introduced AAE module of the architecture (see Fig. 4), after feeding the weighted feature vectors to the encoder, for any time-step t in $[2,T]$, the attention component receives the encoder output $\mathbf{V} = \{\nu_t,\ t \in [1,T]\}$ and the previous hidden state of the decoder h_{t-1}, then computes the attention energy vector e_t, with elements that represent the correlation between the t_{th} frame of the video and the entire set of video frames, using a score function (see below), and finally applies a soft-max function to normalize the attention energies producing the attention weight vector a_t. For $t = 1$, as illustrated in Fig. 4, the attention component uses the hidden state of the last encoder's step ($\mathbf{H_e}$), since there is no previous hidden state of the decoder. Afterwards, the attention weight vector a_t is multiplied (an operation denoted by "MM" in Fig. 4) with the encoder's output, producing the context vector $\nu'_t,\ t \in [1,T]$. The latter is fed to the decoder, which combines it with its output from the previous frame y_{t-1}, so as to incrementally reconstruct the video. The score function used in

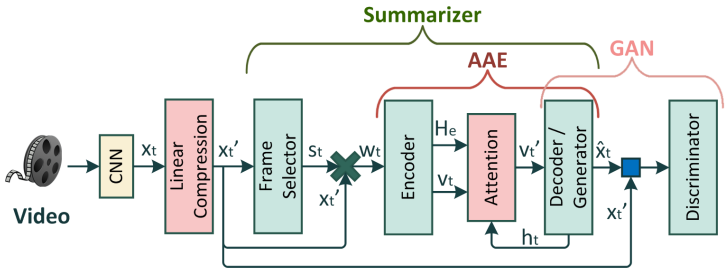

Fig. 3. The proposed SUM-GAN-AAE architecture.

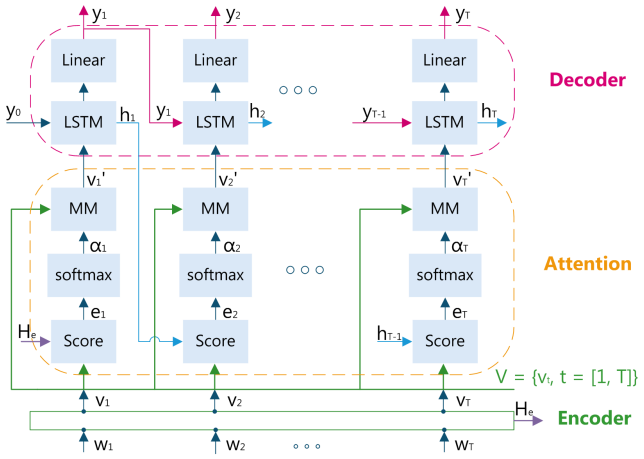

Fig. 4. The attention auto-encoder. Decoding is performed in a stepwise manner which involves the corresponding step of the attention component.

our implementation is a multiplicative one, i.e. $e_t^i = \nu_i^* W_a h_{t-1}$ where ν_i^* is the transposed encoder output for the i_{th} video frame, h_{t-1} is the hidden state of the decoder for $t-1$, W_a is a learnable parameter and e_t^i is the relevance score (scalar value) before the normalization. The final attention weights a_t^i are computed based on the following normalization: $a_t^i = \exp(e_t^i) / \sum_{j=1}^{n} \exp(e_t^j)$.

4 Experiments

4.1 Experimental settings

Datasets. The performance of the developed models is evaluated on the SumMe [10] and TVSum [22] datasets. SumMe includes 25 videos of 1 to 6 min. duration, covering multiple events from both first-person and third-person view. Each video has been annotated by $15-18$ users in the form of key-fragments, and thus is associated to multiple fragment-level user summaries. Moreover, a single ground-truth summary in the form of frame-level importance scores (calculated by averaging the key-fragment user summaries per frame) is also provided. TVSum contains 50 videos of 1 to 5 min. duration, capturing 10 categories of the TRECVid MED dataset. Each video has been annotated by 20 users in the form of frame-level importance scores, and a single ground-truth summary (computed by averaging all users' scores) is available.

Evaluation Approach. For fair comparison with the majority of SoA approaches, we adopt the key-fragment-based evaluation protocol from [27]. Similarity between an automatically created (A) and a user summary (U) is computed by the F-Score (as percentage), where (P)recision and (R)ecall measure the temporal overlap ($\cap$) between the summaries ($||*||$ denotes duration):

$$F = 2 \times \frac{P \times R}{P + R} \times 100, \quad \text{with} \quad P = \frac{A \cap U}{||A||} \quad \text{and} \quad R = \frac{A \cap U}{||U||} \tag{3}$$

So, given a video, we compare the generated summary with the user summaries for this video, and compute an F-Score for each pair of compared summaries. Then, we average the computed F-Scores (for TVSum) or keep the maximum of them (for SumMe, following [11]) and end up with the final F-Score for this video. The computed F-Scores for the entire set of testing videos are finally averaged to capture the algorithm's performance. This protocol is directly applicable on SumMe, as user annotations are already available in the form of key-fragments. For TVSum, frame-level annotations are converted to key-fragment annotations following [22, 27]. The videos are segmented using the KTS method [18], and fragment-level importance scores are computed by averaging the scores of each fragment's frames. Video fragments are ranked based on the computed scores and the Knapsack algorithm is used to select the key-fragments and form the summary, such that it does not exceed 15% of the video's duration (an assumption made by most works in the literature). Finally, for fair comparison with a group of methods ([9, 13, 16, 24, 25, 28]) that evaluate the generated summary for a given video by matching it with the single ground-truth summary for that video, we report our model's performance based on this approach too.

Implementation Details. Videos were downsampled to 2 fps. Feature extraction was based on the pool5 layer of GoogleNet [23] trained on ImageNet. The linear compression layer reduces the size of these vectors from 1024 to 500. Each component of the architecture is comprised of a 2-layer LSTM, with 500 hidden units, while the frame selector is a bi-directional LSTM. Training is based on the Adam optimizer and the learning rate for all components but the discriminator is 10^{-4}; for the latter one is 10^{-5}. Moreover, we followed the typical learning setting (see [27, 16]) where the used dataset is split into two non-overlapping sets; one training set having 80% of data, and one testing set including the remaining 20% of data. Finally, we ran our experiments for 5 different random splits and we report the average performance over these runs.

4.2 Performance evaluation

The developed models were initially evaluated for several values of the regularization factor σ, ranging between 0.05 and 0.5. Greater values were not examined as the models' performance was significantly reduced in (at least) one of the datasets for the highest tested value. In Table 1 we report our findings focusing on the proposed SUM-GAN-AAE model. As can be seen, this factor affects the model's efficiency (as reported in [16]) and thus, it needs fine-tuning. Moreover, the latter seems to be dataset-dependent, as the highest performance is achieved for different values of σ in each dataset. For fair comparison with other methods that use a strictly defined set of parameters, in the following we refer to the SUM-GAN-AAE model with $\sigma = 0.15$, while the best performing SUM-GAN-sl and SUM-GAN-VAAE models were observed for $\sigma = 0.1$ and $\sigma = 0.3$ respectively.

The results of the comparative evaluation of these models against the performance of a randomly generated summary[2] and of other SoA unsupervised

[2]Importance scores were defined based on a uniform distribution of probabilities and the experiment was repeated 100 times.

Table 1. Performance (F-Score (%)) of SUM-GAN-AAE for different values of the regularization factor. Best performance is shown in bold.

	SumMe	TVSum
$\sigma = 0.05$	47.1	58.3
$\sigma = 0.1$	48.2	58.2
$\sigma = 0.15$	**48.9**	58.3
$\sigma = 0.3$	47.6	57.3
$\sigma = 0.5$	46.8	**59.6**

Table 2. Comparison (F-Score (%)) with different unsupervised video summarization approaches, on SumMe and TVSum. +/− indicate better/worse performance than SUM-GAN-AAE.

	SumMe	TVSum
Random summary	39.9 (−)	53.9 (−)
Tessellation [14]	41.4 (−)	**64.1** (+)
DR-DSN [32]	41.4 (−)	57.6 (−)
Online Motion-AE [29]	37.7 (−)	51.5 (−)
UnpairedVSN [20]	47.5 (−)	55.6 (−)
SUM-GAN-sl [1]	47.3 (−)	58.0 (−)
SUM-GAN-VAAE	45.7 (−)	57.6 (−)
SUM-GAN-AAE	**48.9**	58.3

approaches on SumMe and TVSum are reported in Table 2. As a note, each method's score is from the corresponding paper, while the original SUM-GAN method is not listed in this table as it follows a different evaluation protocol; the comparison with it is reported in the sequel (see Tables 4 and 5). These results show that: i) the performance of a few SoA methods is comparable (or even worse) to that of a random summary generator; ii) the best approach on TVSum achieves random-level performance on SumMe, a fact that indicates it is a dataset-tailored technique, since it efficiently summarizes the TVSum videos but clearly fails to define good summaries for the SumMe videos; iii) the introduction of variational attention reduces the efficiency of the SUM-GAN-sl model, possibly due to the difficulty in efficiently defining two latent spaces in parallel to the continuous updating of the model's components during the training; iv) the replacement of the VAE with the AAE results in a noticeable performance improvement over the SUM-GAN-sl model. The latter indicates the contribution of the introduced attention mechanism in enhancing the decoder's ability to identify the most important frames to pay attention to, and in effectively guiding the learning of the adversarial component of the architecture. The applied training strategy efficiently backpropagates this knowledge to the frame selection component, resulting in a significantly improved performance compared to the SUM-GAN-sl model. Moreover, a study of the training curves of the adversarial part of these models (see Fig. 5) points out that the AAE contributes to much faster and more stable training. On top of these findings, the SUM-GAN-AAE model performs consistently well in both datasets (being the best one on SumMe), and thus is the most competitive one among the compared approaches.

Our unsupervised SUM-GAN-AAE model was compared also against supervised methods for video summarization (which is a rather unfair comparison). Table 3 shows that: i) the best methods on TVSum are highly-adapted to this dataset as they exhibit random-level performance on SumMe; ii) only a few supervised methods surpass the performance of a random summary generator on both datasets. The performance of these methods ranges in $44.1 - 49.7$ on SumMe, and in $56.1 - 61.4$ on TVSum. Hence, the results of our unsupervised method make SUM-GAN-AAE comparable with SoA supervised algorithms.

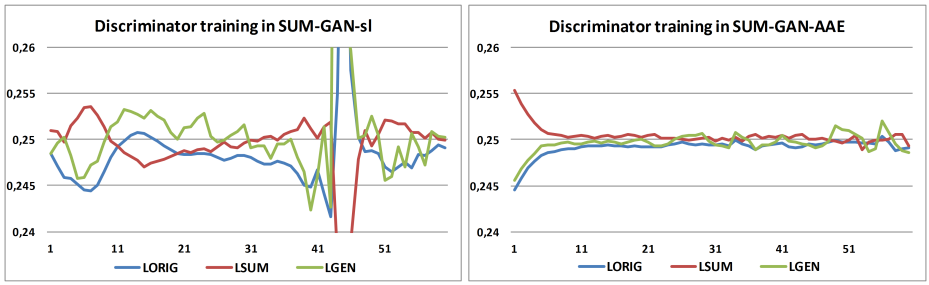

Fig. 5. Loss curves of the discriminator (L_{ORIG}, L_{SUM}) and generator (L_{GEN}) of SUM-GAN-sl and SUM-GAN-AAE models. Horizontal axis denotes training epochs.

Table 3. Comparison (F-Score (%)) of our *unsupervised* method with *supervised* video summarization approaches on SumMe and TVSum. +/− indicate better/worse performance than SUM-GAN-AAE.

	SumMe	TVSum		SumMe	TVSum
Random summary	39.9 (−)	53.9 (−)	MAVS [8]	40.3 (−)	**66.8** (+)
vsLSTM [27]	37.6 (−)	54.2 (−)	SUM-FCN [21]	47.5 (−)	56.8 (−)
dppLSTM [27]	38.6 (−)	54.7 (−)	SUM-DeepLab [21]	48.8 (−)	58.4 (+)
H-RNN [30]	41.1 (−)	57.7 (−)	DR-DSNsup [32]	42.1 (−)	58.1 (−)
Tessellationsup [14]	37.2 (−)	63.4 (+)	ActionRanking [6]	40.1 (−)	56.3 (−)
HSA-RNN [31]	44.1 (−)	59.8 (+)	UnpairedVSNpsup [20]	48.0 (−)	56.1 (−)
DQSN [33]	-	58.6 (+)	VASNet [7]	**49.7** (+)	61.4 (+)
DSSE [26]	-	57.0 (−)	SUM-GAN-AAE	48.9	58.3

For fair comparison with approaches evaluated using the single ground-truth summaries of each video of SumMe and TVSum (i.e. the different evaluation protocol adopted in [9, 13, 16, 24, 25, 28]), we assessed our model also via this approach. Once again we considered different values for the regularization factor σ, to examine its impact on the model's efficiency according to this evaluation protocol and make our findings comparable with the ones in [16]. Table 4 indicates that the model's performance is, indeed, affected by the value of σ, while the effect of this hyper-parameter depends on the evaluation approach (best performance when using multiple human summaries was observed for $\sigma = 0.15$). Moreover, our method clearly outperforms the original SUM-GAN model on both datasets, even for the same value of σ. Finally, the comparison of the best performing instance of our model (for $\sigma = 0.5$) with other techniques that follow this evaluation protocol, indicates the superiority of the proposed approach in both datasets (see Table 5; methods' scores are from the corresponding papers).

5 Conclusions

We presented a video summarization method that combines the effectiveness of attention mechanisms in spotting the most attractive parts of the video and the learning efficiency of the generative adversarial networks for unsupervised training. Based on the SUM-GAN model, we built an improved variation that

Table 4. Comparison (F-Score (%)) of the best performing SUM-GAN model (based on [16]) with the proposed model for different values of the regularization term σ.

	σ	SumMe	TVSum
SUM-GAN	0.3	38.7	50.8
SUM-GAN-AAE	0.05	53.6	60.9
	0.1	55.4	59.9
	0.15	56.4	60.2
	0.3	56.0	59.8
	0.5	**56.9**	**63.9**

Table 5. Comparison (F-Score (%)) of video summarization approaches on SumMe and TVSum, using a single ground-truth summary for each video. Unsupervised methods marked with *. +/− indicate better/worse performance than SUM-GAN-AAE.

	SumMe	TVSum
* SUM-GAN [16]	38.7 (−)	50.8 (−)
* SUM-GANdpp [16]	39.1 (−)	51.7 (−)
SUM-GANsup [16]	41.7 (−)	56.3 (−)
SASUM [24]	45.3 (−)	58.2 (−)
DTR-GAN [28]	44.6 (−)	59.1 (−)
A-AVS [13]	43.9 (−)	59.4 (−)
M-AVS [13]	44.4 (−)	61.0 (−)
AALVS [9]	46.2 (−)	63.6 (−)
* Cycle-SUM [25]	41.9 (−)	57.6 (−)
* SUM-GAN-AAE	**56.9**	**63.9**

performs incremental training of the model's components, applies a stepwise label-based learning approach for the adversarial part and has a reduced number of network parameters. Two further extensions of the developed model were studied; one using a variational attention mechanism and another one using a deterministic attention auto-encoder. Experimental evaluations on the SumMe and TVSum datasets documented the positive contribution of the introduced attention auto-encoder component in the model's training and summarization performance, and highlighted the competitiveness of the proposed unsupervised SUM-GAN-AAE approach against SoA video summarization techniques.

6 Acknowledgments

This work was supported by the EUs Horizon 2020 research and innovation programme under grant agreement H2020-780656 ReTV. The work of Ioannis Patras has been supported by EPSRC under grant No. EP/R026424/1.

References

1. Apostolidis, E., et al.: A stepwise, label-based approach for improving the adversarial training in unsupervised video summarization. In: AI4TV, ACM MM 2019
2. Apostolidis, E., et al.: Fast shot segmentation combining global and local visual descriptors. In: IEEE ICASSP 2014. pp. 6583–6587
3. Apostolidis, K., et al.: A motion-driven approach for fine-grained temporal segmentation of user-generated videos. In: MMM 2018. pp. 29–41
4. Bahuleyan, H., et al.: Variational attention for sequence-to-sequence models. In: 27th COLING. pp. 1672–1682 (2018)
5. Cho, J.: PyTorch implementation of SUM-GAN (2017), https://github.com/j-min/Adversarial_Video_Summary, (last accessed on Oct. 18, 2019)
6. Elfeki, M., et al.: Video summarization via actionness ranking. In: IEEE WACV 2019. pp. 754–763
7. Fajtl, J., et al.: Summarizing videos with attention. In: ACCV 2018. pp. 39–54

8. Feng, L., et al.: Extractive video summarizer with memory augmented neural networks. In: ACM MM 2018. pp. 976–983
9. Fu, T., et al.: Attentive and adversarial learning for video summarization. In: IEEE WACV 2019. pp. 1579–1587
10. Gygli, M., et al.: Creating summaries from user videos. In: ECCV 2014. pp. 505–520
11. Gygli, M., et al.: Video summarization by learning submodular mixtures of objectives. In: IEEE CVPR 2015. pp. 3090–3098
12. Hochreiter, S., et al.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (1997)
13. Ji, Z., et al.: Video summarization with attention-based encoder-decoder networks. IEEE Trans. on Circuits and Systems for Video Technology pp. 1–1 (2019)
14. Kaufman, D., et al.: Temporal Tessellation: A unified approach for video analysis. In: IEEE ICCV 2017. pp. 94–104
15. Lee, S., et al.: A memory network approach for story-based temporal summarization of 360 videos. In: IEEE CVPR 2018. pp. 1410–1419
16. Mahasseni, B., et al.: Unsupervised video summarization with adversarial LSTM networks. In: IEEE CVPR 2017. pp. 2982–2991
17. Otani, M., et al.: Video summarization using deep semantic features. In: ACCV 2016. pp. 361–377
18. Potapov, D., et al.: Category-specific video summarization. In: ECCV 2014. pp. 540–555
19. Radford, A., et al.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR 2016
20. Rochan, M., et al.: Video summarization by learning from unpaired data. In: IEEE CVPR 2019
21. Rochan, M., et al.: Video summarization using fully convolutional sequence networks. In: ECCV 2018. pp. 358–374
22. Song, Y., et al.: TVSum: Summarizing web videos using titles. In: IEEE CVPR 2015. pp. 5179–5187
23. Szegedy, C., et al.: Going deeper with convolutions. In: IEEE CVPR 2015. pp. 1–9
24. Wei, H., et al.: Video summarization via semantic attended networks. In: AAAI 2018. pp. 216–223
25. Yuan, L., et al.: Cycle-SUM: Cycle-consistent adversarial LSTM networks for unsupervised video summarization. In: AAAI 2019. pp. 9143–9150
26. Yuan, Y., et al.: Video summarization by learning deep side semantic embedding. IEEE Trans. on Circuits and Systems for Video Technology **29**(1), 226–237 (2019)
27. Zhang, K., et al.: Video summarization with Long Short-Term Memory. In: ECCV 2016. pp. 766–782
28. Zhang, Y., et al.: DTR-GAN: Dilated temporal relational adversarial network for video summarization. In: ACM TURC 2019. pp. 89:1–89:6
29. Zhang, Y., et al.: Unsupervised object-level video summarization with online motion auto-encoder. Pattern Recognition Letters (2018)
30. Zhao, B., et al.: Hierarchical recurrent neural network for video summarization. In: ACM MM 2017. pp. 863–871
31. Zhao, B., et al.: HSA-RNN: Hierarchical structure-adaptive RNN for video summarization. In: IEEE/CVF CVPR 2018. pp. 7405–7414
32. Zhou, K., et al.: Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In: AAAI 2018. pp. 7582–7589
33. Zhou, K., et al.: Video summarisation by classification with deep reinforcement learning. In: BMVC 2018