A Stepwise, Label-based Approach for Improving the Adversarial Training in Unsupervised Video Summarization

Authors

Evlampios Apostolidis
CERTH-ITI
Thermi, Greece, 57001
Queen Mary University of London
apostolid@iti.gr

Alexandros I. Metsai
CERTH-ITI
Thermi, Greece, 57001
alexmetsai@iti.gr

Eleni Adamantidou
CERTH-ITI
Thermi, Greece, 57001
adamelen@iti.gr

Vasileios Mezaris
CERTH-ITI
Thermi, Greece, 57001
bmezaris@iti.

Ioannis Patras
Queen Mary University of London London,
UK, E14NS
i.patras@qmul.ac.uk

A Stepwise, Label-based Approach for Improving the Adversarial Training in Unsupervised Video Summarization

Name Surname
Institution
Address
E-mail

Name Surname
Institution
Address
E-mail

Name Surname
Institution
Address
E-mail

ABSTRACT

In this paper we present our work on improving the efficiency of adversarial training for unsupervised video summarization. Our starting point is the SUM-GAN model, which creates a representative summary by using it to reconstruct a video that is indistinguishable from the original one. We build on a publicly available implementation of a variation of this model, that includes a linear compression layer to reduce the number of learned parameters and applies an incremental approach for training the different components of the architecture. After assessing the impact of these changes to the model's performance, we propose a stepwise, label-based learning process to improve the training efficiency of the adversarial part of the model. Before evaluating our model's efficiency, we perform a thorough study with respect to the used evaluation protocols and we examine the possible performance on two benchmarking datasets, namely SumMe and TVSum. Experimental evaluations and comparisons with state of the art highlight the competitiveness of the proposed method. An ablation study indicates the benefit of each applied change on the model's performance, and points out the advantageous role of the introduced stepwise, label-based training strategy on the learning efficiency of the adversarial part of the architecture.

CCS CONCEPTS

• **Information systems → Summarization**; **Multimedia content creation**; **Retrieval models and ranking**; • **Computing methodologies → Machine learning**.

KEYWORDS

Video Summarization, Unsupervised Learning, Adversarial Training, Evaluation Protocol, Datasets

ACM Reference Format:
Name Surname, Name Surname, and Name Surname. 2019. A Stepwise, Label-based Approach for Improving the Adversarial Training in Unsupervised Video Summarization. In *Nice '19: 27th ACM International Conference on Multimedia, October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

Nice '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recent advances in video capturing and storage technology and the widespread use of social networks (e.g. Facebook, Twitter), video sharing platforms (e.g. YouTube) and online video archives, facilitated the recording and sharing of huge volumes of video content. Thousands of hours of video are uploaded every single day on the Web, aiming to attract the viewers' attention. Nevertheless, in several cases, browsing through extensive videos to obtain the content that a viewer prefers is a highly time-consuming and tedious process. Hence, the provision of a concise summary that adequately conveys the main concept of the video, enables the viewer to quickly grasp an idea without having to watch the entire content. Given the plethora of videos on the Web and the limited time spent by viewers on deciding whether to see or skip a video, an effective video summary allows time-efficient browsing of large video collections and increases the potential of a video to be consumed.

Video summarization aims to provide a short visual summary that encapsulates the flow of the story and the essential parts of the full-length video. The application domain is widely extended and includes the use of such technologies by video sharing platforms that aim to higher viewer engagement and content consumption, and the content management systems of media organizations to allow effective indexing, browsing and retrieval of video content. Moreover, video summarization that takes into account the diversity of the current content distribution environment, enables effective sharing of video content across different channels (e.g. 4G/5G WANs, local LANs, etc. with various data transmission capacity) and presentation devices (e.g. desktops, laptops, tablets, smart-phones), in forms (storyboards, skims, excerpts) that are tailored to the needs of each viewer, thus facilitating content presentation and consumption.

Several methods aimed to tackle the task of video summarization, and deep learning approaches were the main focus of researchers over the last years. In this direction, a number of datasets were built to facilitate training and evaluation of video summarization algorithms. However, driven by the fact that video summarization is a highly-subjective task, we argue that supervised learning, which relies on the use of a single ground-truth summary, cannot fully explore the potential of deep learning architectures. The latter, in combination with the limited amount of annotated data for training a video summarization algorithm in a supervised manner, directed our focus on improving the performance of an unsupervised method for video summarization. Starting from the work of [16] and building on a PyTorch implementation [3] of a variation of this model, we perform a thorough study with respect to parts and procedures that could be further fine-tuned for improving the models' performance. In particular, after evaluating the

implemented modifications, namely the addition of a linear compression layer that reduces the number of trainable parameters and the application of an incremental training method which updates the model's components in a partial manner, we propose a stepwise, label-based approach for training the adversarial part of the architecture. Experiments on the SumMe and TVSum benchmarking datasets, showed that the proposed method, called "SUM-GAN-sl" in the sequel, exhibits significantly improved performance compared to the original one, and is highly competitive against other state-of-the-art methods. In a nutshell, our contributions include:

- the evaluation of how the variations introduced by the developer of [3], i.e. the addition of the linear compression layer and the applied incremental process for training the architecture, influence the performance of the original model;
- the proposal of a stepwise, label-based approach for training the adversarial module of the network in a more fine-grained manner, and the assessment of the advantage this update brought on the algorithm's efficiency;
- a thorough study of the relevant literature that allowed to gather information about the utilized evaluation protocols and spot the differences in the assessment of state-of-the-art summarization algorithms;
- experiments on the SumMe and TVSum benchmarking datasets, that resulted in estimated regarding the lower and higher bounds of performance and the suitability of the used metrics for evaluating video summarization approaches.

2 RELATED WORK

Several approaches were proposed over the last couple of decades, for addressing the task of video summarization. For the sake of brevity, here we report only on machine learning methods that exploit the learning efficiency of neural networks. A group of algorithms were based on the use of Convolutional Neural Networks (CNN). For example, in [19] video summarization is addressed as a weakly-supervised learning problem and solved via a flexible deep 3D convolutional neural network architecture that learns the notion of importance using only video-level annotation. [23] tackles video summarization as a sequence labeling problem and performs key-frame-based video summarization using fully convolutional sequence models. [6] combines a soft, self-attention network with a 2-layer fully connect network to process the CNN features of the video frames and compute frame-level importance scores that are used for key-fragment selection. [18] uses deep video features for encoding various levels of content semantics and a deep neural network that maps videos and their descriptions to a common semantic space. The latter is jointly trained with associated pairs of videos and descriptions and a summary is created by clustering the extracted deep features from the video segments.

The effectiveness of Recursive Neural Networks (RNN) (e.g. Long Short-Term Memory (LSTM) units [11] and Gated Recurrent Units (GRU) [4]) to capture the temporal dependency over sequential data led to several RNN-based architectures for video summarization that represent the current state-of-the-art. [28] introduces the use of LSTMs to model temporal dependency among frames and compute frame-level importance scores. [31] proposes a 2-layer LSTM architecture where the first layer extracts and encodes data about the video structure and the second layer uses this data to define the key-fragments of the video. This work is extended in [32] to exploit the shot-level temporal structure of the video and compute shot-level confidence scores for producing a key-shot-based summary of the video. [29] describes a Dilated Temporal Relational (DTR) Generative Adversarial Network (GAN), where the generator contains LSTM and DTR units to exploit long-range temporal dependencies at different temporal windows, and the discriminator is trained via a 3-player loss to distinguish between the learned summary and a trivial summary consisting of randomly selected frames. Finally, a number of works focus on introducing attention mechanisms in the network's architecture, to identify the most suitable parts and build the summary e.g. [7, 8, 12].

Besides the aforementioned supervised approaches, a number of unsupervised methods that do not rely their training on annotated data were proposed as well. [16] addresses video summarization by selecting a sparse subset of video frames that optimally represent the input video. A deep summarizer network is trained to minimize the distance between training videos and a distribution of their summarizations through a generative adversarial framework. [33] formulates video summarization as a sequential decision-making process and develops a deep summarization network that learns to produce diverse and representative video summaries via reinforcement learning and a novel reward function. [30] suggests an approach that extracts key motions of appearing objects in the video, and learns to produce a fine-grained object-level video summarization in an unsupervised manner. [23] describes an unsupervised variation of the proposed model, that aims to increase the visual diversity of the selected key-frames. Finally, [22] introduces a new formulation to learn video summarization from unpaired data. Sports highlights, movie trailers and other professionally-edited summary videos available online are collected and used to guide an adversarial process that learns a mapping function of a raw video to a human-like summary.

3 PROPOSED APPROACH

The starting point of our work was the unsupervised method from [16]. The core idea of Mahasseni et al. was to build a keyframe selection mechanism (to generate static video summaries) by minimizing the distance between features extracted from the selected key-frames and the entire video. For this, a deep representation of the entire video frame sequence is created with the help of a bi-directional LSTM, which assigns a weight to each frame, and a variational auto-encoder (VAE). The former is used to capture the long-term dependencies over sequences of frames in both forward and backward direction. The latter is used to reveal the underlying structure of the frame/keyframe features (in its encoding part) and produce another representation of the video by drawing samples from the computed latent space (in its decoding part). The difficulty in defining a suitable threshold regarding the similarity between the reconstructed and the original video, directed Mahasseni et al. to the adversarial framework and the integration of a trainable discriminator network. The ultimate goal of this approach was to jointly train the frame selector and the variational auto-encoder in order to maximally confuse the discriminator, i.e. decrease discriminator's confidence in distinguishing the original from a reconstructed

video, a condition that denotes a highly representative keyframe collection.

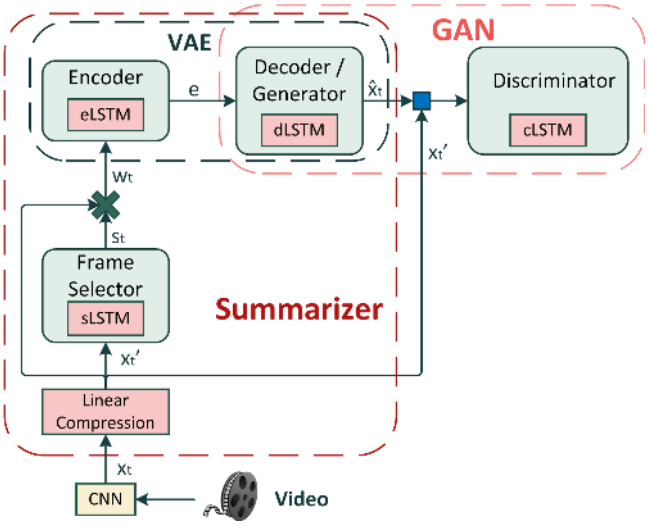

Figure 1: The proposed variation of the SUM-GAN model. The Summarizer (i.e. the Frame Selector (sLSTM), the Encoder (eLSTM) and the Decoder (dLSTM)) has been extended by a linear compression layer that reduces the size of the feature vectors. In addition, the model's components are trained incrementally and the GAN part of the architecture (i.e. the Generator (dLSTM) and the Discriminator (cLSTM)) are trained in a stepwise and label-based manner.

Building on this method, we gained deeper knowledge about the components of the SUM-GAN model and explored the possibility of improving its performance by fine-tuning specific parts of the architecture and the training process. For this, we were based on a publicly available PyTorch implementation of a variation of this architecture [3], that was used for evaluating the performance of SUM-GAN on the summarization of 360° videos (see [15]). This variation contains a linear compression layer right before the frame selection component of the architecture. In the updated model (see Fig. 1), given a video of M frames and focusing on the t^{th} frame of this video, x_t represents the CNN feature vector, x'_t denotes the compressed feature vector, s_t refers to the computed importance score from the frame selector, w_t corresponds to the weighted feature vector ($s_t \otimes x'_t$), and $\hat{x}_t$ relates to the reconstructed feature vector by the variational auto-encoder.

In addition to the added linear layer, this variation follows a 3-step incremental training approach that updates specific parts of the network in each step. In particular, differently to the immediate update of the entire model based on the computed losses after a single forward pass of the architecture (see Alg. 1 in [16]), the implemented process:

- performs a 1^{st} forward pass over the entire model, computes the $L_{reconst}$, L_{prior} and $L_{sparsity}$ losses, and updates only the frame selector, the encoder and the linear compression layer during the 1^{st} backward pass (top part of Fig. 2);
- performs a 2^{nd} forward pass of the partially updated model, computes the $L_{reconst}$ and L_{GAN} losses, and updates only the decoder and the linear compression layer during the 2^{nd} backward pass (middle part of Fig. 2);
- performs a 3^{rd} forward pass of the updated model, computes the L_{GAN} loss, and updates only the discriminator during the 3^{rd} backward pass (bottom part of Fig. 2);

The aforementioned losses are computed similarly to [16]:

$$L_{reconst} = \left\| \varphi(\mathbf{x'}) - \varphi(\hat{\mathbf{x}}) \right\|^2 \quad (1)$$

where $\varphi(\mathbf{x'})$ is the output of the last hidden layer of cLSTM for compressed feature vectors of the original video ($\mathbf{x'} = \{x'_t\}_{t=1}^{M}$) and $\varphi(\hat{\mathbf{x}})$ is the output of the last hidden layer of cLSTM for the feature vectors of the summary-based reconstructed video ($\hat{\mathbf{x}} = \{\hat{x}_t\}_{t=1}^{M}$).

$$L_{prior} = D_{KL}(q(\mathbf{e}|\mathbf{x})||p(\mathbf{e})) \quad (2)$$

where $p(\mathbf{e})$ is a prior over the unobserved latent variable, $\mathbf{x}$ is the observed data, $q(\mathbf{e}|\mathbf{x})$ is the probability of observing $\mathbf{e}$ given $\mathbf{x}$, and D_{KL} denotes the Kullback-Leibler divergence. For efficient training we employ the re-parameterization trick proposed in [14].

$$L_{sparsity} = \left\| \frac{1}{M} \sum_{t=1}^{M} s_t - \sigma \right\|^2 \quad (3)$$

where M is the total number of video frames and σ is the regularization factor, a tunable hyper-parameter of the model.

$$L_{GAN} = log(cLSTM(\mathbf{x'})) + log(1 - cLSTM(\hat{\mathbf{x}})) + log(1 - cLSTM(\hat{\mathbf{x}}_\mathbf{p})) \quad (4)$$

where $cLSTM(\mathbf{x'})$, $cLSTM(\hat{\mathbf{x}})$ and $cLSTM(\hat{\mathbf{x}}_\mathbf{p})$ are probability scores (computed at the soft-max output of the discriminator) representing the discriminator's confidence when classifying the original video, the generated summary and the uniform summary respectively.

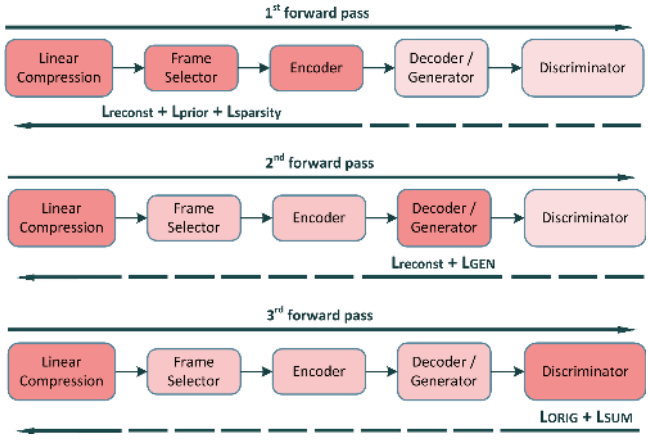

Figure 2: The different parts of the architecture are trained through a 3-step, incremental procedure that updates specific components of the model in each step. Solid line and dark-coloured boxes indicate the updated parts of the architecture during the backward pass. Dashed line and light-coloured boxes correspond to the unaltered components of the model during the backward pass.

Given the above, we examined a different training strategy for the adversarial part of the model. The introduced learning approach was utilized in [21] for unsupervised representation learning with deep convolutional GANs, a method used for image generation. Driven by the effectiveness of this approach on training a network to generate realistic images from white noise, we transfer this methodology in our context. Our aim is to a find a better equilibrium point between the generator and the discriminator, which means a better reconstruction of the video from the combination of the weighted frames and the learned distribution of data by the variational auto-encoder of the architecture. So, instead of using the L_{GAN} loss of the original SUM-GAN model, we follow a label-based approach, were label "1" is assigned to the original video and label "0" to the video summary. Given these labels, we introduce the following two losses:

$$L_{ORIG} = (1 - cLSTM(\mathbf{x'}))^2 \; and \; L_{SUM} = (cLSTM(\hat{\mathbf{x}}))^2 \quad (5)$$

The L_{ORIG} is used to minimize the Mean Square Error (MSE) between the original video label and the computed probability when the discriminator is fed with the original video. Respectively, the L_{SUM} is used to minimize the MSE between the summary label and the computed probability when the discriminator is fed with the summary-based reconstruction of the video. Based on these losses, the training of the discriminator is performed in a stepwise manner, as depicted in Fig. 3 (top part). First, we pass the compressed feature vectors of the original video ($x'_t,\ t = [1, M]$) through the discriminator (forward pass), calculate L_{ORIG} and then calculate the gradients (backward pass). Secondly, we pass the original video through the summarizer to create the reconstructed video ($\hat{x}_t,\ t = [1, M]$), forward the latter to the discriminator, calculate L_{SUM} and then accumulate the gradients from both the original video and the summary-based reconstructed one, with another backward pass. With the gradients accumulated, we call a step of the discriminator's optimizer. This incremental process enables a more fine-grained computation of the discriminator's gradients (compared with the training policy used in SUM-GAN), and helps the discriminator develop higher discrimination efficiency, thus performing better during the classification.

For training the generator, we introduce the following loss:

$$L_{GEN} = (1 - cLSTM(\hat{\mathbf{x}}))^2 \quad (6)$$

The L_{GEN} is used to minimize the MSE between the original video label and the computed probability when the discriminator is fed with the summary-based reconstruction of the video. By constantly trying to reduce the sum of $L_{Reconst}$ and L_{GEN}, the generator aims to confuse the discriminator and make the summary-based reconstruction of the video indistinguishable from the original one.

The reasoning behind choosing the MSE loss instead of the commonly used Binary Cross Entropy (BCE) loss for training the GAN module of the architecture, resides in the fact that in vanilla GANs, the latent vector (random noise) is sampled independently of the training data. The original GAN has shown better performance with the BCE Loss, since it does not force the network to learn a non-meaningful representation between the noise vector and a ground-truth image. Instead, it helps the generator to produce more versatile outputs, taking into account only if the output is passed as

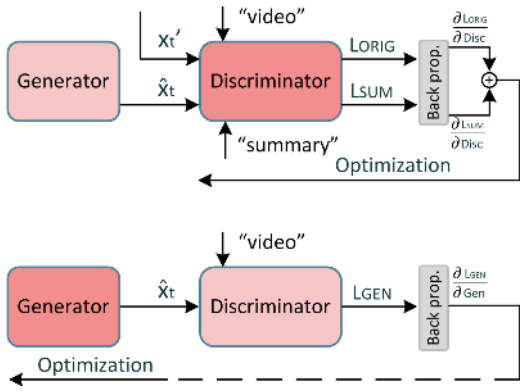

Figure 3: The stepwise, label-based training of the adversarial component of our model. Top part corresponds to the Discriminator and bottom part to the Generator.

real or fake. In our method, differently from original GANs, the introduction of a variational auto-encoder alters the above approach due to the input no longer being a random noise latent vector, but an original video to be reconstructed and fed to the discriminator for comparison. Therefore, we choose the MSE as the loss function, since our method attempts to reconstruct the input and to not generate new samples. To validate this choice we performed a set of experiments and their findings are reported in Section 4.

Given the above described training strategy, the randomly generated summary used in the original SUM-GAN model to regularize learning of the discriminator is not needed any more in our variation. The authors of [16] claim that the use of the randomly generated summary enhances the discriminator's ability to distinguish between the original video and a summary-based reconstruction of it. Nevertheless, through this approach the discriminator learns to classify the random summary in the same class with the generated summary, thus restricting the discriminator's ability to make the distinction between an actual video summary and a randomly generated one. Based on this reasoning, we omit the use of a random summary for training our model.

After training, the components responsible for generating a summary for an unseen video are the linear compression layer and the frame selector. In particular, the CNN features of the video frames pass through the aforementioned components and an importance score is computed for each frame. Based on these scores, the key-fragments of the video are selected via the following procedure: the video is segmented using the KTS algorithm [20] (other approaches for shot or subshot segmentation, e.g. [1] and [2], could be used too); then, fragment-level importance scores are calculated by averaging the importance scores of each fragment's frames; and finally, the summary is generated by selecting the fragments that maximize the total importance score provided that the length of the summary does not exceed 15% of the original video duration, a convention adopted by most video summarization approaches. This latter step is performed by solving the following optimization problem:

$$max \sum_{i=1}^{N} a_i \cdot b_i,\ s.t. \sum_{i=1}^{N} a_i \cdot l_i \leq 0.15 \cdot L,\ a_i \in 0, 1 \quad (7)$$

where N is the number of fragments, L is the length of the original video, 0.15 defines the upper limit for the summary duration, and given the $i-th$ fragment of the video, a_i is a binary value that indicates whether the fragment is selected or not, b_i is the computed fragment-level importance score, and l_i is the length of the fragment. The latter is the 0/1 Knapsack problem.

4 EXPERIMENTS

4.1 Datasets

We evaluate the performance of our method on SumMe [10] and TVSum [24]. The former includes 25 videos covering multiple events from both first-person and third-person view, while the video duration ranges from 1 to 6 minutes. The latter contains 50 videos capturing 10 categories of the TRECVid Multimedia Event Detection dataset and the length of each video ranges from 1 to 5 minutes. In terms of ground-truth annotation, each video of SumMe has been annotated by 15 to 18 viewers/users in the form of key-fragments, and thus it is associated to multiple fragment-level user summaries. Moreover, besides the aforementioned user summaries, a single ground-truth summary in the form of frame-level importance scores (calculated as an average of the key-fragment user summaries per frame) is also provided. In the case of TVSum, videos have been annotated by 20 viewers/users in the form of frame-level importance scores. Similar to SumMe, a single ground-truth summary in the form of frame-level importance scores (computed after averaging all users' scores) is provided for each video of the dataset.

4.2 Evaluation Approach

For fair comparison with other video summarization algorithms, we adopt the evaluation protocol proposed in [28]. The similarity between a generated summary and a ground-truth summary is computed by the harmonic mean of Precision and Recall and expressed as the F-score in percentages.

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100 \tag{8}$$

Precision and Recall express frame-level temporal overlap between the generated (S) and the ground-truth (G) summary:

$$Precision = \frac{Duration\ of\ overlap\ between\ S\ and\ G}{Duration\ of\ S} \tag{9}$$

$$Recall = \frac{Duration\ of\ overlap\ between\ S\ and\ G}{Duration\ of\ G} \tag{10}$$

A thorough study of the relevant literature indicated that the vast majority of works evaluate the performance of video summarization based on the key-fragment [1] protocol introduced in [28]. As stated before, the ground-truth annotations for the SumMe dataset are already available in the form of key-fragments which can be used directly for evaluation. Nevertheless, the annotation of the TVSum videos is available only in the form of frame-level importance scores. To tackle this, the frame-level ground-truth annotations of the TVSum videos are converted to key-fragment-based summaries following the approach presented in [24, 28]. In particular, the videos are temporally segmented into non-overlapping fragments using the KTS method [20]. Then, fragment-level importance scores are computed by averaging the importance score of the frames of each fragment, and the calculated scores are used for ranking the fragments. Finally, a subset of fragments is selected to form the video summary, such that the summary duration does not exceed 15% of the video duration. In most cases, the latter is performed using the Knapsack algorithm, as proposed in [24, 28].

Given the above technical background, we found out that there is a slight but significant distinction with respect to what is eventually used as ground-truth summary for evaluating the performance of a video summarization algorithm. In particular, a number of works (see Tables 4 and 5) compare the generated summary for a given video against the single ground-truth summary that is available for that video in SumMe and TVSum (mainly for supervised training). Differently to this approach, a larger group of works (see Table 7) evaluate the efficiency of the generated summary for a given video by assessing its similarity with all available human-generated summaries for that video. Driven by the fact that video summarization is a highly subjective task, we argue that exploiting existing knowledge from many human-generated summaries can lead to more concrete and reliable results. Hence, in our assessments we follow the evaluation protocol that involves all human-generated summaries. More precisely, given video, we compare the generated summary with the available user summaries and compute an F-score for each pair of generated and user summary. Then, we average the computed F-scores (in the case of TVSum) or keep the maximum of them (in the case SumMe, following the recommendation of the authors of this dataset (see [9])) and end up to the final F-score for this video. The computed F-scores for the entire set of testing samples, are finally averaged and form the final outcome about the algorithms performance. For fair comparison with methods that adopt the single ground-truth summary evaluation approach, we report our model's performance based on this approach too.

4.3 Preliminary Study on Datasets

Aiming to get some insights about the used datasets, we examined the following aspects:

- the efficiency of a randomly generated summary (frames' importance scores defined based on a uniform distribution of possibilities and the experiment was performed 100 times);
- the human performance, i.e. how well a human annotator would perform based on the preferences of the remaining annotators; this is a metric regarding the compatibility/agreement between the defined human summaries;
- an estimate about the highest performance on TVSum [2] according to the best human-generated summary (with the highest overlap) for each video of the dataset.

For completeness, in Table 1 we report the outcomes of our study using both criteria for calculating the video-level F-scores, i.e. the maximum of the computed F-scores in the case of SumMe, and the average of these scores in the case of TVSum. The results - which are consistent with the findings of a recently published study on these datasets [17] - clearly indicate that video summarization is a highly subjective task, as there is no ideal summary that exhibits significant overlap with all annotators' preferences, in

[1] Also mentioned as keyshot evaluation protocol in other works.

[2] Based on the "max" criterion, the upper-bound for SumMe is 100%, i.e. the generated summary perfectly matches with a human-generated summary.

both datasets. Moreover, the "average" metric in the case of TVSum shows that human performance is comparable with the efficiency of a randomly generated summary, and thus limits the available space for improvement. In particular, the best possible summary (i.e. a summary that matches the best human-generated summary for each different video of the dataset) results in a score that is approximately 10 units higher than the score of a random summary. Given the reasonable lack of an objective summary for a video, we argue that the "max" criterion is more suitable for assessing the performance of video summarization approaches. In this sense, the upper-bound with respect to video summarization efficiency will be 100% in both datasets, denoting that machine-generated summaries are indistinguishable from human-generated ones.

Table 1: Findings on the performance of different types of summaries and the theoretical upper-bound of the SumMe and TVSum dataset, based on the "average" and "max" criterion. Values denote F-score %.

	SumMe		TVSum	
	Average	Max	Average	Max
Random	18.1	39.9	53.9	75.5
Human Summaries	31.3	55.1	53.8	77.5
Best Possible	44.7	100.0	64.7	100.0

4.4 Implementation Details

We downsampled all videos of the SumMe and TVSum datasets to 2 fps. For fair comparison with several works (including [16]), we use the output of pool5 layer of GoogleNet [25] trained on ImageNet, for representing the visual content of the video frames. The extracted feature vectors contain 1024 elements. The linear compression layer reduces the size of these feature vectors to 500, and this is the number of hidden units of each LSTM layer of the proposed architecture, while all LSTM modules are comprised of two layers. Similarly to [16], the frame selection LSTM is a bi-directional one. Training is based on the Adam optimizer and the learning rate for all components but the discriminator is 10^{-4}; for the latter one equals to 10^{-5}. Finally, we follow the standard 5-fold cross validation approach, i.e. 80% of videos used for training and the rest 20% for testing. Hence, in the following section we report the average performance over the 5 runs.

4.5 Performance Evaluation

The performance of the proposed variation of the SUM-GAN model was initially evaluated for several values of the regularization factor σ. The lower bound for this hyperparameter was set to 0.05 and the higher was set to 0.5. Experiments for greater values were omitted as the method's performance was reduced for the higher tested value. The results reported in Table 2, indicate that: i) the regularization factor clearly affects the performance (as also reported in [16]) and thus needs fine-tuning; ii) too small and too big values lead to reduced efficiency, and only a specific range of values results in good performance; iii) fine-tuning of σ seems to be dataset-dependent, as the highest performance for the model, is achieved for different values in each dataset.

Table 2: Performance of the proposed model for different values of the regularization term σ. Best performance highlighted in bold. Values denote F-score %.

	SumMe	TVSum
σ = 0.05	44.7	58.2
σ = 0.1	**47.3**	58.0
σ = 0.15	46.6	58.6
σ = 0.3	46.4	**58.8**
σ = 0.5	42.7	58.6

For fair comparison with other video summarization methods that rely on a strictly defined set of (hyper-)parameters, in the following we refer to our model with $\sigma = 0.1$, since the gain compared to the model's performance in SumMe for $\sigma = 0.3$, is higher than the observed mitigation in TVSum for $\sigma = 0.1$. The training curves of this model for 100 epochs of training on SumMe and TVSum, are illustrated in Figs. 4 and 5 respectively. In both cases the model starts from approx. the performance of a randomly-generated summary and develops knowledge about the task (the fluctuation in the case of SumMe is reasonable due to the adversarial nature of the training), which results in a noticeable improvement of its summarization efficiency. The pick value was observed in epoch 93 for SumMe and in epoch 98 for TVSum.

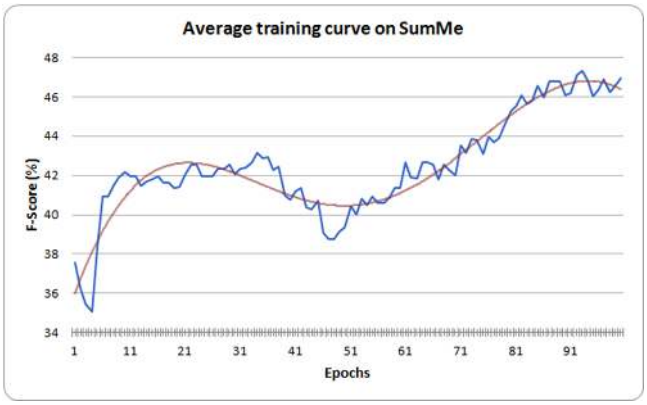

Figure 4: In blue, the average (over 5 splits) training curve of the proposed variation on SumMe. In red, the computed 6-order polynomial that approximates the training curve.

Before delving into more details with respect to the conducted comparisons with the current state of the art, in Table 3 we present our findings regarding the effect of the selected criterion for training the GAN part of the architecture, on the model's performance. The replacement of the MSE by the BCE loss led to a noticeable decrease in the algorithm's efficiency on SumMe, while maintained its performance on TVSum. Therefore, it seems that the use of the MSE loss can be beneficial in the case of limited training data (for SumMe we used 20 training samples), enabling the model to converge in a state that achieves higher performance. The results for TVSum indicate that both criteria result to similar efficiency on

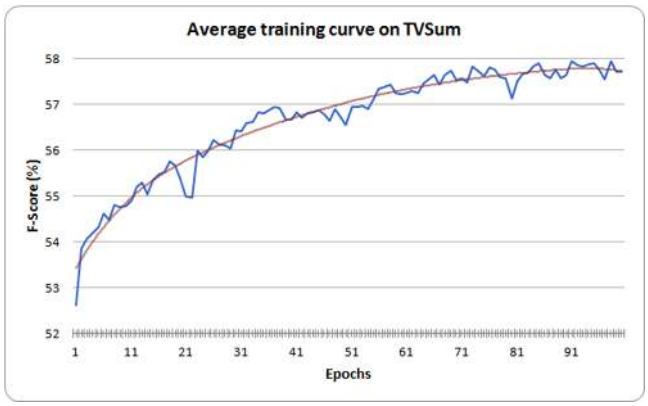

Figure 5: In blue, the average (over 5 splits) training curve of the proposed variation on TVSum. In red, the computed 6-order polynomial that approximates the training curve.

larger sets of training samples (for TVSum we used 40 training samples), that allow the GAN to be updated with similar effectiveness over the training epochs, in both cases.

Table 3: Findings regarding the effectiveness of each training criterion for the GAN part of the architecture, on the SumMe and TVSum datasets. Values denote F-score %.

	SumMe	TVSum
MSE Loss	47.3	58.0
BCE Loss	44.6	58.0

Our model was compared against other state-of-the-art unsupervised approaches on SumMe and TVSum. The reported data in Table 4 [3], point out that: i) the performance of a group of algorithms is comparable (or even worse) than the efficiency of a randomly generated summary; ii) the best method on SumMe (UnpairedVSN) performs slightly better than our method, while it is clearly less competitive on TVSum; iii) the best algorithm on TVSum (Tessellation) achieves random-level performance on SumMe, a fact that indicates a dataset-tailored technique. Contrary to the above, our approach performs consistently well in both datasets, thus being the most competitive one among the compared techniques.

Furthermore, the efficiency of our unsupervised method was compared against the performance of supervised approaches for video summarization. From the data presented in Table 5 it is shown that: i) the two best methods in TVSum (MAVS and Tessellation-sup respectively) are highly-adapted to this dataset, as they exhibit random-level performance on SumMe; ii) only a few methods clearly surpass the performance of a randomly-generated summary on both datasets, with VASNet being the best among them. The performance of the latter methods ranges from 44.1 to 49.7 in SumMe, and from 56.1 to 61.4 on TVSum. Hence, the performance of our SUM-GAN-sl model (47.3 on SumMe and 58.0 on TVSum) makes our unsupervised method comparable with state-of-the-art supervised techniques for video summarization.

[3]The scores for each method are from the corresponding paper.
[4]Performance reported in a subsequent work of the authors (see [32]).

Table 4: Performance evaluation of different unsupervised video summarization approaches on SumMe and TVSum, taking under consideration all human-generated summaries for each video. Symbols (+), (-) indicate better, worse result than that of the proposed SUM-GAN-sl. Values denote F-score %.

	SumMe	TVSum
Random	39.9 (-)	53.9 (-)
DR-DSN [33]	41.4 (-)	57.6 (-)
UnpairedVSN [22]	47.5 (+)	55.6 (-)
Tessellation [13]	41.4 (-)	64.1 (+)
Online Motion-AE [30]	37.7 (-)	51.5 (-)
SUM-GAN-sl	47.3	58.0

Table 5: Comparison of our *unsupervised* method with *supervised* video summarization approaches on SumMe and TVSum, after taking under consideration all human-generated summaries for each video. Symbols (+), (-) indicate better, worse result than that of the proposed SUM-GAN-sl. Values denote F-score %.

	SumMe	TVSum
Random	39.9 (-)	53.9 (-)
vsLSTM [28]	37.6 (-)	54.2 (-)
dppLSTM [28]	38.6 (-)	54.7 (-)
H-RNN [31] [4]	41.1 (-)	57.7 (-)
HSA-RNN [32]	44.1 (-)	59.8 (+)
DQSN [34]	-	58.6 (+)
DSSE [27]	-	57.0 (-)
VASNet [6]	49.7 (+)	61.4 (+)
MAVS [7]	40.3 (-)	66.8 (+)
SUM-FCN [23]	47.5 (+)	56.8 (-)
SUM-DeepLab [23]	48.8 (+)	58.4 (+)
ActionRanking [5]	40.1 (-)	56.3 (-)
DR-DSNsup [33]	42.1 (-)	58.1 (+)
UnpairedVSNpsup [22]	48.0 (+)	56.1 (-)
Tessellationsup [13]	37.2 (-)	63.4 (+)
SUM-GAN-sl	47.3	58.0

Finally, for fair comparison with works that rely their evaluations on the single ground-truth summaries of each video of SumMe and TVSum, we assessed the performance of our method also via this approach. As a preliminary experiment, we examined different values for the regularization factor σ, to check the consistency of our findings with what has been discussed in [16]. The reportings in Table 6 indicate that: i) the method's performance is affected by the modification of σ in a way similar to the one reported in [16]; ii) the effect of this hyperparameter strongly depends on the used evaluation approach (best performance when using multiple human summaries was observed for $\sigma = 0.1$); and iii) our method clearly outperforms the original SUM-GAN model on both datasets, even for the same value of σ. The comparison of the best performing version of our model (for $\sigma = 0.5$) with other summarization techniques (both supervised and unsupervised ones) that follow

this evaluation protocol, indicated the superiority of the proposed approach in both benchmarking datasets (see Table 7 [3]).

Table 6: Comparison of the best performing SUM-GAN model (based on the score reported in [16]) with the performance of the proposed model for different values of the regularization term σ. Values denote F-score %.

	SumMe	TVSum
Original SUM-GAN (σ = 0.3)	38.7	50.8
SUM-GAN-sl (σ = 0.1)	38.1	61.0
SUM-GAN-sl (σ = 0.3)	45.2	62.4
SUM-GAN-sl (σ = 0.5)	**46.8**	**65.3**

Table 7: Performance evaluation of different video summarization approaches on SumMe and TVSum, using a single ground-truth summary for each video. Unsupervised methods are in *italics*. Symbols (+), (-) indicate better, worse result than that of the proposed SUM-GAN-sl. Values denote F-score %.

	SumMe	TVSum
SUM-GAN [16]	38.7 (-)	50.8 (-)
SUM-GANdpp [16]	39.1 (-)	51.7 (-)
SUM-GANsup [16]	41.7 (-)	56.3 (-)
A-AVS [12]	43.9 (-)	59.4 (-)
M-AVS [12]	44.4 (-)	61.0 (-)
SASUM [26]	45.3 (-)	58.2 (-)
DTR-GAN [29]	44.6 (-)	59.1 (-)
SUM-GAN-sl	46.8	65.3

4.6 Ablation Study

To see how each introduced change influences the performance of the proposed model we conducted an ablation study. The variations taken under consideration, as well as their performance on SumMe and TVSum, are reported in Table 8. From these values it seems that: i) the replacement of the incremental training of the architecture, by the sequential one described in [16] leads to a significant performance reduction on SumMe and a slight decrement on TVSum (see Var. 3); ii) a similar effect is observed with respect to the linear compression layer (see Var. 2), as its removal results in a bit lower performance (compared to Var. 3) in both datasets; iii) the addition of the linear compression layer and the application of the incremental training for the model's components (see Var. 1) led to a clear performance improvement in SumMe (more than 2%) and a slight amelioration in TVSum (reaching 0.5% compared to Var. 2); iv) the introduction of the stepwise, label-based training strategy for the GAN module of the architecture, advanced further the model's performance on SumMe (by 0.8%) and maintained the same efficiency on TVSum.

The above indicate that the incremental training approach is beneficial in case of small training datasets, while its contribution is less pronounced in case of larger datasets. Similarly, the addition of a linear layer that significantly reduces the amount of trained parameters advances the model's training capacity in case of small training sets (as for SumMe), while a lower impact is observed in case of larger training sets (as for TVSum). A possible justification for the above findings is that the amount of training samples in the case of TVSum is adequate for learning a larger set of parameters and through 1-step training. The application of the stepwise, label-based learning approach enables the adversarial part of the model to converge to a more ideal state through a more fine-grained update of the discriminator's gradients and the use of a more strictly defined learning task for the generator. This strategy seems to be profitable in the case of small training sets, while it maintains the same levels of (state-of-the-art) performance when larger groups of training samples are used. To sum up, the applied changes contributed to significantly improve the performance of the original SUM-GAN model, and the introduced GAN-training approach allowed the model to reach higher levels of performance on SumMe, making it comparable with the best performing unsupervised method.

Table 8: Ablation study based on performance evaluation of three variations of the proposed model on SumMe and TVSum. Values denote F-score %.

	SUM-GAN-sl	Var. 1	Var. 2	Var. 3
Incremental training	✓	✓	✓	X
Linear compression	✓	✓	X	✓
Stepwise GAN train	✓	X	✓	✓
Performance on SumMe & TVSum	**47.3** & 58.0	46.5 & 58.0	44.0 & 57.5	44.3 & 57.9

5 CONCLUSIONS AND NEXT STEPS

This paper reported our study for assessing and advancing the effectiveness of a unsupervised video summarization method that is based on adversarial learning. Focusing on the SUM-GAN model and after assessing the efficiency of a variation of it, we suggested a new training approach to advance the learning efficiency of the adversarial module of the architecture. A thorough study of the evaluation protocols and metrics, and experiments on two datasets allowed to estimate the possible performance on these datasets and the suitability of the used metrics. Comparative evaluations showed that our model performs constantly well on both datasets and is among the best unsupervised methods, while its efficiency make it comparable with supervised algorithms too. An ablation study proved the contribution of each applied change and the gain offered by the proposed stepwise, label-based adversarial training strategy. In the future we plan to put effort on further improving our model, e.g. by exploiting the efficiency of attention networks and the training capacity of reinforcement learning approaches, and we will investigate approaches for video summarization that is tailored to targeted audience and the used distribution channel.

6 ACKNOWLEDGMENTS

This work was supported by the EUs Horizon 2020 research and innovation programme under grant agreement H2020-780656 ReTV.

REFERENCES

[1] Evlampios Apostolidis and Vasileios Mezaris. 2014. Fast shot segmentation combining global and local visual descriptors. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6583–6587.

[2] Konstantinos Apostolidis, Evlampios Apostolidis, and Vasileios Mezaris. 2018. A Motion-Driven Approach for Fine-Grained Temporal Segmentation of User-Generated Videos. In *MultiMedia Modeling*, Klaus Schoeffmann, Thanarat H. Chalidabhongse, Chong Wah Ngo, Supavadee Aramvith, Noel E. O'Connor, Yo-Sung Ho, Moncef Gabbouj, and Ahmed Elgammal (Eds.). Springer International Publishing, Cham, 29–41.

[3] Jaemin Cho. 2017. *PyTorch Implementation of SUM-GAN from "Unsupervised Video Summarization with Adversarial LSTM Networks".* https://github.com/j-min/Adversarial_Video_Summary (last accessed on July 10, 2019).

[4] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179

[5] Mohamed Elfeki and Ali Borji. 2019. Video Summarization Via Actionness Ranking. In *IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, January 7-11, 2019*. 754–763. https://doi.org/10.1109/WACV.2019.00085

[6] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. 2019. Summarizing Videos with Attention. In *Computer Vision – ACCV 2018 Workshops*, Gustavo Carneiro and Shaodi You (Eds.). Springer International Publishing, Cham, 39–54.

[7] Litong Feng, Ziyin Li, Zhanghui Kuang, and Wei Zhang. 2018. Extractive Video Summarizer with Memory Augmented Neural Networks. In *Proceedings of the 26th ACM International Conference on Multimedia (MM '18)*. ACM, New York, NY, USA, 976–983. https://doi.org/10.1145/3240508.3240651

[8] Tsu-Jui Fu, Shao-Heng Tai, and Hwann-Tzong Chen. 2019. Attentive and Adversarial Learning for Video Summarization. In *IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, January 7-11, 2019*. 1579–1587. https://doi.org/10.1109/WACV.2019.00173

[9] Michael Gygli, Helmut Grabner, and Luc Van Gool. 2015. Video summarization by learning submodular mixtures of objectives. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3090–3098. https://doi.org/10.1109/CVPR.2015.7298928

[10] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. 2014. Creating Summaries from User Videos. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 505–520.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[12] Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. 2019. Video Summarization with Attention-Based Encoder-Decoder Networks. *IEEE Transactions on Circuits and Systems for Video Technology* (2019), 1–1. https://doi.org/10.1109/TCSVT.2019.2904996

[13] Dotan Kaufman, Ggil Levi, Tal Hassner, and Lior Wolf. 2017. Temporal Tessellation: A Unified Approach for Video Analysis. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 94–104. https://doi.org/10.1109/ICCV.2017.20

[14] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. http://arxiv.org/abs/1312.6114

[15] Sangho Lee, Jinyoung Sung, Youngjae Yu, and Gunhee Kim. 2018. A Memory Network Approach for Story-based Temporal Summarization of 360 Videos. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[16] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. 2017. Unsupervised Video Summarization with Adversarial LSTM Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2982–2991.

[17] Mayu Otani, Yuta Nakahima, Esa Rahtu, and Janne Heikkilä. 2019. Rethinking the Evaluation of Video Summaries. In *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[18] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. 2016. Video Summarization using Deep Semantic Features. In *ACCV*.

[19] Rameswar Panda, Abir Das, Ziyan Wu, Jan Ernst, and Amit K. Roy-Chowdhury. 2017. Weakly Supervised Summarization of Web Videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 3677–3686. https://doi.org/10.1109/ICCV.2017.395

[20] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. 2014. Category-Specific Video Summarization. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 540–555.

[21] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *2016 International Conference on Learning Representations (ICLR)*.

[22] Mrigank Rochan and Yang Wang. 2019. Video Summarization by Learning From Unpaired Data. In *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[23] Mrigank Rochan, Linwei Ye, and Yang Wang. 2018. Video Summarization Using Fully Convolutional Sequence Networks. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 358–374.

[24] Yale Song, J. Vallmitjana, A. Stent, and A. Jaimes. 2015. TVSum: Summarizing web videos using titles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5179–5187. https://doi.org/10.1109/CVPR.2015.7299154

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1–9. https://doi.org/10.1109/CVPR.2015.7298594

[26] Huawei Wei, Bingbing Ni, Yichao Yan, Huanyu Yu, Xiaokang Yang, and Chen Yao. 2018. Video Summarization via Semantic Attended Networks. In *2018 AAAI Conference on Artificial Intelligence (AAAI)*.

[27] Yitian Yuan, Tao Mei, Peng Cui, and Wenwu Zhu. 2018. Video Summarization by Learning Deep Side Semantic Embedding. *IEEE Transactions on Circuits and Systems for Video Technology* (2018), 1–1. https://doi.org/10.1109/TCSVT.2017.2771247

[28] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video Summarization with Long Short-Term Memory. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 766–782.

[29] Yujia Zhang, Michael Kampffmeyer, Xiaodan Liang, Dingwen Zhang, Min Tan, and Eric P. Xing. 2018. DTR-GAN: Dilated Temporal Relational Adversarial Network for Video Summarization. *CoRR* abs/1804.11228 (2018).

[30] Yujia Zhang, Xiaodan Liang, Dingwen Zhang, Min Tan, and Eric P. Xing. 2018. Unsupervised object-level video summarization with online motion auto-encoder. *Pattern Recognition Letters* (2018). https://doi.org/10.1016/j.patrec.2018.07.030

[31] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. 2017. Hierarchical Recurrent Neural Network for Video Summarization. In *Proceedings of the 2017 ACM on Multimedia Conference (MM '17)*. ACM, New York, NY, USA, 863–871. https://doi.org/10.1145/3123266.3123328

[32] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. 2018. HSA-RNN: Hierarchical Structure-Adaptive RNN for Video Summarization. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*.

[33] Kaiyang Zhou and Yu Qiao. 2018. Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity-Representativeness Reward. In *2018 AAAI Conference on Artificial Intelligence (AAAI)*.

[34] Kaiyang Zhou, Tao Xiang, and Andrea Cavallaro. 2018. Video Summarisation by Classification with Deep Reinforcement Learning. In *2018 British Machine Vision Conference (BMVC)*.