Video Summarization with Attention-Based Encoder-Decoder Networks

Zhong Ji[1], Kailin Xiong[1], Yanwei Pang[1], and Xuelong Li[2]
[1]Tianjin University, Tianjin, China
[2]Xi'an Institute of Optics and Precision Mechanics, Xi'an, China

***Abstract*—This paper addresses the problem of supervised video summarization by formulating it as a sequence-to-sequence learning problem, where the input is a sequence of original video frames, the output is a keyshot sequence. Our key idea is to learn a deep summarization network with attention mechanism to mimic the way of selecting the keyshots of human. To this end, we propose a novel video summarization framework named *Attentive encoder-decoder networks for Video Summarization* (AVS), in which the encoder uses a *Bidirectional Long Short-Term Memory* (BiLSTM) to encode the contextual information among the input video frames. As for the decoder, two attention-based LSTM networks are explored by using additive and multiplicative objective functions, respectively. Extensive experiments are conducted on three video summarization benchmark datasets, i.e., SumMe, TVSum, and YouTube. The results demonstrate the superiority of the proposed AVS-based approaches against the state-of-the-art approaches, with remarkable improvements from 3% to 11% on the three datasets, respectively.**

***Index Terms*—Video summarization, LSTM, encoder-decoder, attention mechanism.**

I. Introduction

VIDEO is inundating the Internet social platform. There are more than 300 hours video upload per minute to YouTube. It is awfully time-consuming to browse these videos. According to Ciscos 2015 Visual Networking Index, it will take over 500 million years to watch all videos uploaded to Internet per month in the year of 2020! It is therefore becoming increasingly important to efficiently browse, manage, and retrieve these videos.

Video summarization is one of the promising techniques to address this challenge [1]–[6]. Its goal is to produce a compact yet comprehensive summary to enable an efficient browsing experience. An ideal video summarization is that can provide users the maximum information of the target video with the shortest time. It is also useful for many other practical applications, such as video indexing [7], video retrieval [8], and event detection [9].

Generally, there are two types of video summarization: storyboard and video skim. Specifically, a storyboard is based on a set of keyframes, and a video skim is composed of a number of representative video segments, called keyshots. In this work, we focus on video skim. However, it can be easily converted to the form of storyboard by selecting one or several keyframes from each keyshot.

Video summarization has been studied over two decades [1]–[6]. During these years, many approaches have been developed by exploring cues ranging from low-level visual inconsistency [10] [11], attention [3] [5] [26], to high-level semantic change of concepts [6] [13] and entities in videos [12] [34]. However, most of these studies focus on unsupervised leaning technique. Recently, the research focus has been extending to supervised learning approaches [14]–[19], which aims at explicitly learning the summarizing capability from the human labels. Usually, supervised approaches have better performance than unsupervised ones.

Among the previous supervised approaches, studies in [17] and [18] are attractive ones. They treat video summarization as a sequence-to-sequence learning problem, where the input is the original video frame sequence and the output is the keyframe/keyshot sequence. To obtain a good video summarization, the complex and heterogeneous inter-dependency should be well considered. Both studies explore the encoder-decoder framework with *Long Short-Term Memory* (LSTM) technique to model the variable-range dependencies in video summarization. As a specific type of *Recurrent Neural Network* (RNN), LSTM has shown its effectiveness in modeling long-range dependencies where the influence by the distant states on the present and future states can be adaptively adjusted and data-dependent [20]. Therefore, both [17] and [18] achieve state-of-the-art performances.

However, one main drawback in such an encoder-decoder framework [17] [18] is that it encodes all the necessary information in one single context vector no matter how long the input sequence is. Thus, the length of the intermediate code is fixed in their encoder-decoder models, which incapacitates it to give different weights to different frames in the input sequence explicitly. In this situation, all the shots/frames in the input video sequence have the same importance no matter what kind of output shots/frames are to be predicted. Due to this indiscriminate averaging of all the frames, both approaches [17] [18] risks ignoring much of the temporal structure underlying the video. For example, considering summarizing a video "leave home to walk dog and then come back". Since the video frames related to the "home scene" are visually similar, it is hard for both approaches to tell the order of appearances from the collapsed vectors.

To this end, we explore the attentive encoder-decoder framework to tackle this problem in video summarization. The framework employs attention mechanism in the encoder-decoder framework by conditioning the generative process in the decoder on the encoder hidden states, rather than on one single context vector only. We name this framework *Attentive encoder-decoder networks for Video Summarization* (AVS). In specific, we use attention mechanism [21] [22] in the AVS framework, which can assign importance weights to different shots/frames of the input instead of treating all the input ones equally. In this way, it provides the inherent relations between the input video sequence and the output keyshots. Figure 1 shows an overview of AVS framework. Compared with previous work, this paper has several essential characteristics worth being highlighted:

1) It proposes an *Attentive encoder-decoder framework for Video Summarization*, named AVS. It is a supervised-based video summarization framework, which mimics the way of selecting the keyshots of human. To the best of our knowledge, this attentive encoder-decoder framework has not previously proposed for implementing video summarization.

2) It investigates the attention-based LSTM mechanism in the AVS framework, and develops two approaches to generate the video summarization. One is based on additive attention mechanism named A-AVS, the other is based on multiplicative attention mechanism, named M-AVS.

3) Extensive experiments are conducted on three popular video summarization datasets, including both the edited and raw video datasets. The results show the proposed M-AVS and A-AVS approaches consistently outperform the state-of-the-art ones by at least 10.9%, 3.1%, and 3.3% on SumMe, TVSum, and Youtube datasets, respectively. These promising results verify the effectiveness of the proposed AVS framework.

The remainder of this paper is organized as follows. Section II reviews the related video summarization methods. Section III introduces the proposed AVS framework and two specific approaches. Section IV presents the experimental results and analysis. Finally, conclusions and future work are provided in Section V.

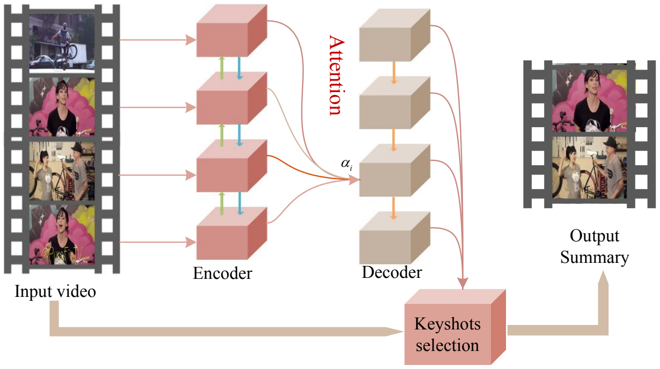


Fig. 1. An overview of the proposed AVS framework. It includes an encoder-decoder model and a keyshot selection model. The encoder first reads the sequence of frames and then the attention based decoder generates a sequence of importance scores. Finally, the keyshot selection model generates the keyshots based on the visual sequence and the output of the decoder.

II. Related Work

According to the number of videos to be summarized, there are *Single-Video Summarization* (SVS) and *Multi-Video Summarization* (MVS). Specifically, SVS aims at digesting one individually long video [27]–[30], while MVS aims at summarizing a large number of short videos obtained by a query to web videos [4] [31] [32] [37]. MVS may also called query-based video summarization, and its processing method is generally different from that of SVS since it has to handle diverse query-based videos and can take advantage of the query information. Furthermore, there is a direction of studying multi-view video summarization, which mainly used in surveillance scenarios to compact the videos captured from different cameras [40]. In our work, we focus on SVS.

From the perspective of the learning model, there are unsupervised and supervised video summarization approaches. In the following, we will introduce their related work in detail. In particular, our work is a supervised approach. Additionally, since our work applies attention-based LSTM network and LSTM is a special type of RNN, we will further review the existing RNN-based and attention-based video summarization approaches, respectively.

A. Unsupervised and Supervised Video Summarization

Unsupervised approaches dominate the field of video summarization for a long time. They are generally designed to make the summarization meets the desired properties, such as conciseness, representativeness, and informativeness. Thus, the corresponding selection criteria for summaries include content frequency [27] [28], coverage [29] [30], relevance [4] [24] [25], and user's attention [3] [26], etc. According to these different criteria, numerous approaches have been developed. Among them, clustering-based methods are the most popular ones [27] [28]. It clusters the visually similar frames or shots into groups, in which the group centers are considered as the representative elements of the video and therefore selected as the keyframes or keyshots. Dictionary learning is another popular technique used in unsupervised video summarization [29] [30]. It regards the base vectors in the dictionary model as the keyframes or keyshots since they can maximally reconstruct the visual content of the original video.

Recently, supervised video summarization approach has also received much research focus. It takes videos and their human-labeled summaries as training data to seek supervised learning methods to explicitly learn how human would summarize videos. For example, Gong *et al.* [14] treat video summarization as a supervised subset selection problem, and present a probabilistic model called sequential *Determinatal Point Process* (seqDPP) to learn how a diverse and representative subset is selected from the training set. Potapov *et al.* [24] train a set of SVM classifiers to score each segment in a video with importance score, and those segments with higher scores constitute a video summary.

Besides, some work tend to directly optimize the multiple objectives for video summarization. For instance, Gygli *et al.* [15] learn to combine the criteria of representative, relevance, and uniformity to ensure the generated summaries are the most consistent with the reference ones. Specifically, they develop several submodular functions for these criteria and learn a linear combination of them using structured learning with a large margin formulation. Similarly, Li *et al.* [19] design four functions for the criteria of representativeness, importance, diversity and storyness, respectively. And then, they build a score function to linearly combine the four functions with the maximum margin algorithm. Particularly, their proposed framework is general for both edited and raw video summarization. More recently, some deep architectures with RNN network for supervised video summarization have also been proposed [17] [18], which will be introduced in the next sub-section in detail.

B. RNN-Based Video Summarization Approaches

To the best of our knowledge, the only existing RNN-based video summarization approaches are [17] and [18]. In [17], video summarization is considered as a structured prediction problem on sequential data, and a bidirectional LSTM is used to model the variable-range dependency in the video. The method is called vsLSTM. Specifically, its input is a sequence of video frames and its output is a binary indicator vector (being selected or not) or frame-level importance scores. To enhance the diversity, the authors further introduce *Determinatal Point Process* (DPP) algorithm to vsLSTM, which is called dppLSTM. In [18], an unsupervised generative adversarial learning model is presented, which is called SUM-GAN. Specifically, the generator is an autoencoder LSTM. Its goal is to select video frames and decode the obtained summarization for reconstructing the input video. In contrast, the discriminator is another LSTM network aiming at distinguishing between the original video and its reconstruction from the generator. Furthermore, the authors also extend SUM-GAN method to a supervised setting by adding a sparse regularization with the ground-truth summarization labels, the corresponding method is named SUM-GAN$_{\text{sup}}$. Both methods achieve the state-of-the-art performances in the field of video summarization. In this paper, we treat video summarization as a sequential encoder-decoder problem, and formulate it with an attention-based LSTM framework.

Video highlight [23] and storyline [32] have similar goals to video summarization, thus we also give brief views for existing methods using RNN network in both directions. Specifically, video highlight is a moment of major or special interest in a video. Yang *et al.* [23] cast it as an outlier detection problem where the non-highlights are considered as outliers. Then, they apply recurrent autoencoder with LSTM cells to model temporal dependencies to identify video highlights. Sigurdsson *et al.* [32] propose a *Skipping Recurrent Neural Network* (S-RNN) to learn a storyline from a photo stream. The goal of storyline is to learn the underlying visual appearances and temporal dynamics simultaneously when given hundreds of albums for a concept. Specifically, S-RNN skips through the photo sequences to extract the common latent stories.

C. Attention-Based Video Summarization Approaches

Users attention implies the concentration of mental powers upon a video segment [3] [5] [26]. If a video segment captures much attention of a user, it is more important and more likely to be a keyshot. Existing methods usually apply low-level features, such as motion and face to score the importance of video segments by modeling the users attention. These scores join together to form an attention cue, and those on the curve crests are extracted as the keyshots to construct the summarization.

For example, Ma *et al.* [3] present a set of attention models via multiple sensory perceptions, such as motion, static, face, camera attention, and audio saliency. Then, these models are fused linearly and nonlinearly, respectively. Ejaz *et al.* [26] explore the static attention by using the image signature based saliency detection method, and model the dynamic attention with temporal gradients. Then, they combine both attention models non-linearly to build video summarization. Ngo *et al.* [10] represent a video with a temporal graph of scenes, shots and sub-shots, where motion-based attention values are attached to each node. By modeling the evolution of a video through the temporal graph, the scene changes can be detected and the summary can be generated. More recently, to reduce the computational cost on computing the attention clues, Zhang *et al.* [5] propose a simple but effective motion state change model by using a spatiotemporal slice to analyze the attention curve.

Although these attention modeling schemes have proved to be effective in video summarization, there are still some drawbacks. On the one hand, the attention curve is usually constructed with one or several low-level features. However, one feature cannot well reflect the users attention, and several features cannot typically guarantee a correlation with what the user is interested in [35]. On the other hand, due to the unsupervised characteristics, the existing attention-based approaches cannot take advantage of human guidance. In contrast, our proposed AVS framework can well utilize this guidance since it learns the attention mechanism in a supervised manner. Moreover, its deep neural network framework also guarantee it can capture the complex attention mechanism of viewers.

III. The Proposed AVS Framework

We formulate video summarization as a sequence-to-sequence learning problem, where the input is a sequence of video frames , and the output is a sequence of keyshot. The flowchart of AVS framework is illustrated in Fig. 1. It consists of two components: an encoder-decoder model and a keyshot selection model. Particularly, the encoder-decoder model consists of an encoder and a decoder. It measures the importance of each frame. The key shots selection model aims at converting the frame-level importance scores into shot-level scores and generating summary with a length budget [25].

In this section, we first introduce the encoder network with a bidirectional LSTM, and then present the decoder network with attention mechanism, finally introduce the keyshot selection model briefly.

A. Encoder with Bidirectional LSTM Network

In a common encoder-decoder framework, an encoder converts the input sequence $X = \{x_1, x_2, ..., x_T\}$ into a representation vector $\mathbf{v}$= $\{v_1,v_2,\cdots,v_{\mathrm{T}}\}$.

$$\begin{bmatrix} v_t \\ h_t \end{bmatrix} = \phi(x_t), \tag{1}$$

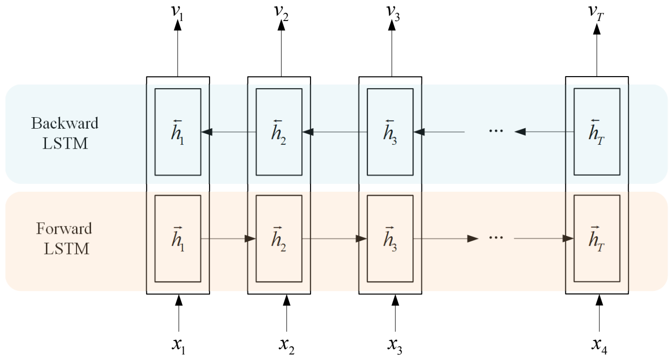


Fig. 2. Illustration of the BiLSTM in the proposed AVS framework.

where $h_t \in \mathbb{R}^n$ is a hidden state at time t. The architecture of an encoder ϕ depends on the input in a specific application. For instance, in the application of image caption [33], *Convolutional Neural Network* (CNN) is a good choice. In the case of machine translation [21] [22], it is natural to use a RNN as the encoder, since its input is a variable-length sequence of symbols. When applied to video summarization, LSTM is the most suitable algorithm [17] [18] since the contextual information around a specific frame is necessary for generating a video summary. It is because human relies on high-level semantic understanding of the video contents, usually after viewing the whole sequence can she/he decide which frame or shot should be selected into the summary. For example, considering summarizing a basketball game video, only a key ball that affects the game process should be selected into the summary. However, there are many goals in a basketball game, thus it is necessary to combine the scene before and after the goal to determine whether a goal is a key ball.

Inspired by outstanding performance of *Bidirectional Long Short-term Memory* (BiLSTM) to encode the necessary information in a sequence [38], we select it as an encoder for taking the temporal relation of video frames into consideration. The principle of BiLSTM is to split the neurons of a regular LSTM [30] into two directions, one for positive time direction (forward states), and the other for negative time direction (backward states). Moreover, those two states outputs are not connected. By utilizing the two-time directions, the sequential information from the past and future of the current frame can be used.

The flowchart of BiLSTM is shown in the encoder part of Fig. 2. First, the forward LSTM reads the input sequence in its forward direction (from x_1 to x_T) and calculates the forward hidden states ($\overrightarrow{h}_1, \cdots, \overrightarrow{h}_T$). Meanwhile, the backward LSTM reads the sequence in the reverse order, resulting in a sequence of backward hidden states($\overleftarrow{h}_1, \cdots, \overleftarrow{h}_T$). Then we obtain an annotation v_t for each x_t by concatenating the forward hidden state $\overrightarrow{h}_t$ and the backward one $\overleftarrow{h}_t$. That is to say, the annotation v_t incorporates the information of both the preceding frames and the following frames. Due to the time tendency of an LSTM, the annotation v_t can focus on the frames around x_t.

B. Decoder with Attention Mechanism

A decoder generates the corresponding output sequence $Y = \{y_1, \cdots, y_m\}$ with the representation vector from the encoder. Similar to that in the encoder, the architecture of the decoder ψ is determined by the output in a specific application. In the application of video summarization, LSTM is the preferred decoder model since it runs sequentially over the output sequence [18]. Generally speaking, there is a contextual relationship for each frame in a video. Due to the importance scores among frames are basically continuous in a video shot and varied among the shots, a decoder should learn the long term and short term dependency among these scores. An LSTM decoder can be written as:

$$\begin{bmatrix} p(y_t|\{y_i|i<t\}, \mathbf{v}) \\ s_t \end{bmatrix} = \psi(s_{t-1}, y_{t-1}, \mathbf{v}). \tag{2}$$

However, the representation vector $\mathbf{v}$ in Eq. (2) is a fixed length encoding vector and cannot accurately describe the temporal characteristics of a video. To exploit the temporal ordering across the entire video, we introduce attention mechanism [21] [22] to it. Then the decoder can be changed as:

$$\mathrm{V}_t = \sum_{i=1}^{n} \alpha_t^i v_i, s.t. \sum_{i=1}^{n} \alpha_t^i = 1, \tag{3}$$

$$\begin{bmatrix} p(y_t|\{y_i|i<t\}, V_t) \\ s_t \end{bmatrix} = \psi(s_{t-1}, y_{t-1}, V_t), \tag{4}$$

where V_t stands for the attention vector at moment t. The attention weight α_t^i is a parameter to trade-off the inputs and the encoder vector. The attention mechanism allows the decoder to selectively focus on only a subset of inputs by increasing their attention weights. The attention mechanism in the LSTM decoder is shown in Fig. 3. The attention weight α_t^i is computed at each time step t , and it reflects the attention degree of the i-th temporal feature in the input video. To obtain α_t^i , the relevance score e_t^i should be computed. This is because that it combines the previous hidden state s_{t-1} in the LSTM decoder and the output of the encoder at time step i . The

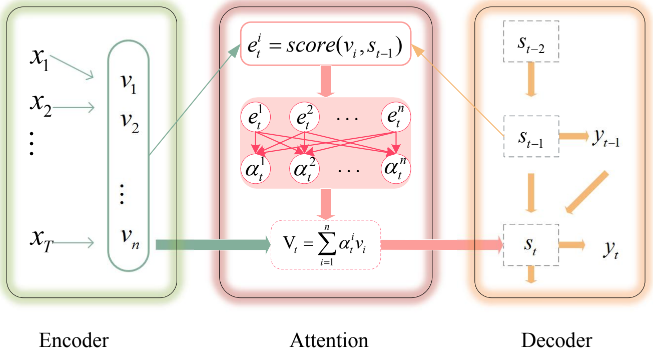


Fig. 3. Illustration of the proposed attention mechanism in the LSTM decoder. To generate decoder output y_t at time t, a score function first combines the i-th encoder output v_i and the last hidden state of the decoder s_{t-1} to obtain the relevance score e_t^i. Second, e_t^i is normalized to gain the attention weight α_t^i. Finally, the decoder input is obtained by weighted sum.

score function that computes the relevance score e_t^i can be written as:

$$e_t^i = score(s_{t-1}, v_i). \tag{5}$$

The score function in Eq. (5) decides the relationship between the i-th visual features v_i and the output scores at time t. It can be implemented in variable ways. Concretely, we develop two models: A-AVS and M-AVS, respectively. As shown in Fig. 4.(a), the A-AVS model applies an additive score function:

$$e_t^i = w^T \tanh(w_a s_{t-1} + U_a v_i + b_a), \tag{6}$$

where w, w_a, U_a are the weights of the additive score function and b_a is the bias. These parameters are estimated together with all other parameters of the encoder and decoder networks. The A-AVS model simply concatenates the video frames and the hidden states of the decoder. Considering a special condition that the outputs of the decoder and visual frames are matched in video summarization. That is to say, a video frame feature v_i corresponds to the hidden state s_{t-1} of the decoder. However, the additive function does not take full advantage of this relationship. To take a better use of the relationship between the outputs of the decoder and the visual frames, we further present an M-AVS model by exploring a multiplicative score function.

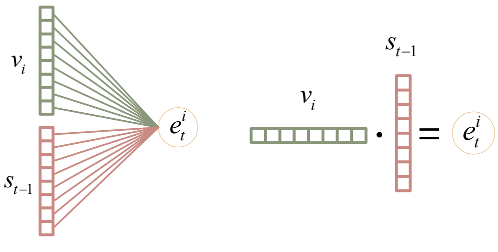
$$e_t^i = v_i^T W_a s_{t-1}. \tag{7}$$

M-AVS model is shown in Fig. 4. (b).

Once the relevance socres e_t^i for all frames $i = 1, \cdots, n$ are computed, we normalize them to obtain the α_t^i by:

$$\alpha_t^i = \exp(e_t^i) / \sum_{j=1}^{n} \exp(e_t^j). \tag{8}$$

Intuitively, this implements an attention mechanism in the decoder. The decoder decides which parts of the source frames to pay attention to. Then the importance score of each frame can be computed.



(a)A-AVS (b)M-AVS

Fig. 4. Illustration of the proposed score functions in the A-AVS and M-AVS model to score the relationship between the input and the output, where v_i represents the i-th vector encoded by the encoder and s_{t-1} stands for the hidden state of the decoder at time $t-1$.

C. Keyshots Selection

Once obtained the predicted importance scores for all frames, the remaining work is to select the keyshots to generate the video summarization. Specifically, we apply the *Kernel Temporal Segmentation* (KTS) proposed by Potapov *et al.* [24] to segment the visually coherent frames into shots. Then it computes shot-level importance scores by taking an average of the frame importance scores within each shot. To generate keyshot-based summary, we need to solve the following optimization problem:

$$\max \sum_{i=1}^{m} u_i w_i, \quad \text{s.t.} \sum_{i=1}^{m} u_i l_i \le 1, u_i \in \{0, 1\}\ , \tag{9}$$

where s is the number of shots, w_i is the importance score of the i-th shot, and l_i is the length of the i-th shot. Note that this is exactly the 0/1 knapsack problem, which can be solved by the dynamic programming method [25]. The summary is then created by concatenating those shots with $u_i \neq 0$ in a chronological order.

IV. Experiments and Analysis

This section first introduces the implementation details, including the datasets, evaluation metrics, and experimental settings. Then, we provide the main experimental results and parameter analysis. Next, we provide additional experiments with data argumentation. Finally, qualitative results are provided.

A. Implementation Details

1) Datasets: We evaluate the proposed AVS framework on three publicly available benchmark datasets: SumMe [16], TVSum [25], and Youtube [27]. Most of the videos in these datasets are 1 to 10 minutes in length. Specifically, SumMe [16] consists of 25 raw videos recording a variety of events such as holidays and sports. TVSum [25] contains 50 edited videos downloaded from YouTube in 10 categories, such as changing vehicle tire, getting vehicle unstuck, grooming an animal. The video contents in both datasets are diverse and include both ego-centric and third-person camera. In addition, both of SumMe and TVSum datasets provide frame-level importance scores for each video, which are used as the ground-truth labels. YouTube [27] contains 50 videos selected from *Open Video Project* (OVP) [36]. The video contents include cartoons, news and sports. This dataset provides multiple user-annotated subsets of keyframes for each video, and we follow the standard approach described in [17] to create a single ground truth set for evaluation. For all the three datasets, we follow the steps in [17] to convert frame level scores to keyshot summaries. Table I summarizes the key characteristics of these datasets.

2) Evaluation Metrics: We apply the popular F-measure as the evaluation metric [15]–[19]. Similar to [17] and [18], our methods generate a summary S which is less than 15% in duration of the original. Given a generated summary S and the ground-truth summary G, we compute the precision P and the recall R for each pair of S and G based on the temporal overlaps between them, as follows:

$$P = \frac{\text{overlaped duration of } S \text{ and } G}{\text{duration of } S}, \tag{10}$$

TABLE I
DESCRIPTIVE STATISTICS OF THE THREE DATASETS.

Dataset	#Video	Descriptions	Duration(Min)	Annotations
SumMe [16]	25	User generated videos of events	1.5–6.5	Frame-level importance scores
TVSum [25]	50	Edited videos (10 categories)	1–5	Frame-level importance scores
YouTube [27]	50	Web videos (sports, news, etc)	1–10	Selected keyframes

TABLE II
PERFORMANCE COMPARISON (F-SCORE) WITH STATE-OF-THE-ART METHODS. BEST RESULTS ARE DENOTED IN **BOLD**.

Dataset	Method	Feature	Supervised/unsupervised	F-score
SumMe	SUM-GAN$_{\text{dpp}}$ [18]	GoogleNet	unsupervised	39.1
	Gygli *et al.* [15]	DeCAF	supervised	39.7
	Zhang *et al.* [41]	AlexNet	supervised	40.9
	vsLSTM [17]	GoogleNet	supervised	37.6
	dppLSTM [17]	GoogleNet	supervised	38.6
	SUM-GAN$_{\text{sup}}$ [18]	GoogleNet	supervised	41.7
	Li *et al.* [19]	VGGNet-16	supervised	43.1
	A-AVS(ours)	GoogleNet	supervised	54.0
	M-AVS(ours)	GoogleNet	supervised	**54.3**
TVSum	TVSum [25]	HoG+GIST+SIFT	unsupervised	51.3
	SUM-GAN$_{\text{dpp}}$ [18]	GoogleNet	unsupervised	51.7
	vsLSTM [17]	GoogleNet	supervised	54.2
	dppLSTM [17]	GoogleNet	supervised	54.7
	SUM-GAN$_{\text{sup}}$ [18]	GoogleNet	supervised	56.3
	Li *et al.* [19]	VGGNet-16	supervised	52.7
	A-AVS(ours)	GoogleNet	supervised	59.4
	M-AVS(ours)	GoogleNet	supervised	**61.0**
Youtube	SUM-GAN$_{\text{dpp}}$ [18]	GoogleNet	unsupervised	60.1
	SUM-GAN$_{\text{sup}}$ [18]	GoogleNet	supervised	62.5
	Zhang *et al.* [41]	GoogleNet	supervised	61.0
	A-AVS(ours)	GoogleNet	supervised	65.8
	M-AVS(ours)	GoogleNet	supervised	**66.2**

$$R = \frac{\text{overlaped duration of } S \text{ and } G}{\text{duration of } G}. \tag{11}$$

Finally, the F-measure is computed as:

$$F = \frac{2 \times P \times R}{(P + R)} \times 100\%. \tag{12}$$

3) Experimental Settings: We downsample the videos into frame sequences in 2 fps. For fair comparison with [17] and [18], we choose to use the output of pool5 layer of the GoogLeNet [39] (1024 dimensionality), trained on ImageNet, as the visual feature for each video frame. Both proposed models have three LSTM layers, and each layer contains 256 units. The attention scale of the decoder is set as 9. As for the training/testing data, we apply the same standard supervised learning setting as [17] [18] where the training and testing are from the disjoint part of the same dataset. We randomly leave 20% for testing and the remaining 80% for training.

To learn parameters in the LSTM layers, we use annotations in the forms of the frame-level importance scores. For both A-AVS and M-AVS, we stop training after 5 consecutive epochs with descending summarization F-score. We set attention scales to 9. For fair comparison, we run both A-AVS and M-AVS for 10 times and report the average performance.

B. Comparison and Analysis

1) Comparison with State-of-the-art Approaches: Eight state-of-the-art video summarization approaches are selected for comparison with our AVS framework, including both unsupervised and supervised approaches. The performance results of the selected approaches are all from the original papers. Particularly, we are interested in comparing our performance in contrast with prior supervised approaches within the deep encoder-decoder framework, i.e., vsLSTM [17], dppLSTM [17], and SUM-GAN$_{\text{sup}}$ [18]. We also choose three additional supervised approaches for comparison.The first one is Li *et al.* [19], which is a general framework designed for both edited and raw videos with the idea of property-weight learning. The second one is Gygli *et al.* [15], which learns submodular mixtures of objectives for different criteria directly. The third one is Zhang *et al.* [41], which learns nonparametrically to transfer summary structures from training videos to test ones. Moreover, two unsupervised approaches, SUM-GAN$_{\text{dpp}}$ [18] and TVSum [25] are chosen for comparison.

Table II shows the comparison results. We can observe that both A-AVS and M-AVS clearly outperform all the competitors in all the datasets. Specifically, on SumMe dataset, our approaches outperform the others in over 11 absolute points. On the other two datasets, there are at least 3 absolute points better than the state-of-the-arts. The significant improvements on SumMe against TVSum and Youtube mainly lies in the fact that SumMe dataset is characterized by slowly changing shots and few scenes. Thus, it is more suitable for attention mechanism to focus on the most important part of a video, which leads to a better performance on SumMe dataset.

In addition, it can be seen that the M-AVS model performs better than the A-AVS model on the three benchmark datasets in about 0.3%-0.6%. This is mainly due to that the multiplicative score layer makes better use of the relationship between the hidden states of the decoder and the visual feature than the additive score one. Even A-AVS has inferior performance, it outperforms SUM-GAN$_{dpp}$, the prior best method with deep encoder-decoder framework, in 11.3%, 3.1%, and 3.3% on SumMe, TVSum and Youtube datasets, respectively. The promising results prove the effectiveness and superiority of our proposed AVS framework.

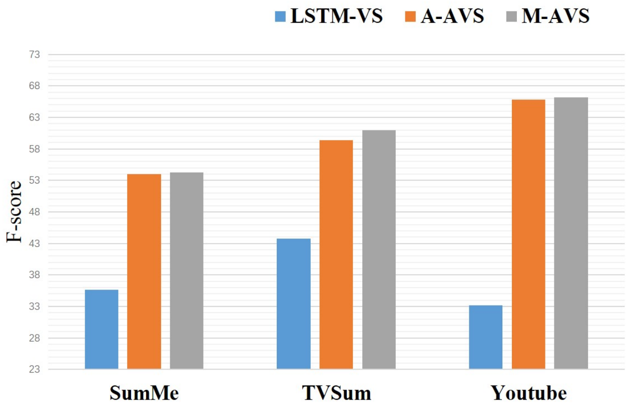


Fig. 5. Comparison of the proposed methods with/without attention mechanism.

2) Importance Evaluation of Attention Mechanism: To better verify the effectiveness of the attention mechanism in AVS framework, we abandon the attention layer in AVS to build a baseline named LSTM-VS. Figure 5 illustrates the performance comparison. It is clear to see that AVS framework outperforms the non-attention based LSTM-VS model noticeably (15%-30%), which also demonstrates the effectiveness of attention mechanism.

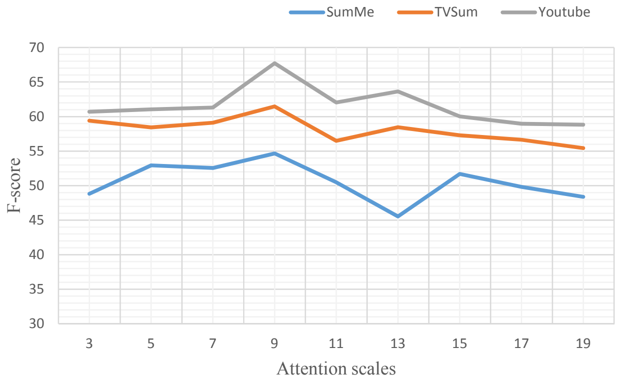


Fig. 6. F-score results of A-AVS model for different values of attention scales on SumMe, TVSum, and Youtube datasets, respectively.

3) Parameter Sensitive Analysis: We evaluate the performances of our methods with different attention scales. Figure 6 shows the F-score values on three different datasets. It can be seen that the performances reach their peaks when the attention scale is around 9. It is maybe due to the fact that each shot is around 9 frames on average when we perform KTS to segment the video into shots. Therefore, we can conclude that the proposed methods will perform better when their attention scales are close to the length of shots.

TABLE III
SUMMARIZATION RESULTS (F-SCORE) WITH OUR AVS FRAMEWORK IN THE AUGMENTED SETTING. BEST RESULTS ARE DENOTED IN **BOLD.**

Dataset	Method	Canonical	Augmented
SumMe	dppLSTM [17]	38.6	42.9
	SUM-GAN$_{sup}$ [18]	41.7	43.6
	A-AVS(ours)	54.0	55.8
	M-AVS(ours)	54.3	**56.1**
TVSum	dppLSTM [17]	54.7	59.6
	SUM-GAN$_{sup}$ [18]	56.3	61.2
	A-AVS(ours)	59.4	60.8
	M-AVS(ours)	61.0	**61.8**

C. Augmentation Experiments

Zhang *et al.* [17] and Mahasseni *et al.* [18] augment the SumMe and TVSum datasets with OVP [36] and YouTube datasets to further improve the performance on SumMe and TVSum. Following their settings, we implement the augmented experiments in AVS framework. Particularly, for a given dataset, we randomly leave 20% of it for testing and augment the remaining 80% with the other three datasets to form an augmented training dataset. The results in Table III clearly indicates that augmenting the training dataset with annotated data from other datasets improves summarization performance. For SumMe, the performances of both proposed methods rise about 1.8%. For TVSum, the performance of A-AVS method has been improved by 1.3%, while that of M-AVS method has been slightly improved by 0.8%. Moreover, the augmented performances for both datasets outperform the comparative approaches. These results confirm that our models are still effective and competitive when performing data augmentation.

D. Qualitative Results

To better illustrate the temporal selection pattern of different variations of our approach, we demonstrate the selected frames on an example video in Fig. 7. It shows the results from vsLSTM, LSTM-VS, A-AVS, and M-AVS models on the 48-th video of the TVSum dataset. The ground-truth frame-level importance scores of the video are represented by the blue blocks. The marked orange intervals are the ones selected by vsLSTM, LSTM-VS, A-AVS, and M-AVS model respectively. We can see that the summaries generated by our methods are more uniform distribution in time than that generated by vsLSTM model. Besides, our A-AVS and M-AVS approaches select more shots with larger importance scores than the others.

V. CONCLUSIONS AND FUTURE WORK

We propose a deep attentive framework for supervised video summarization. Specifically, two attention-based deep models

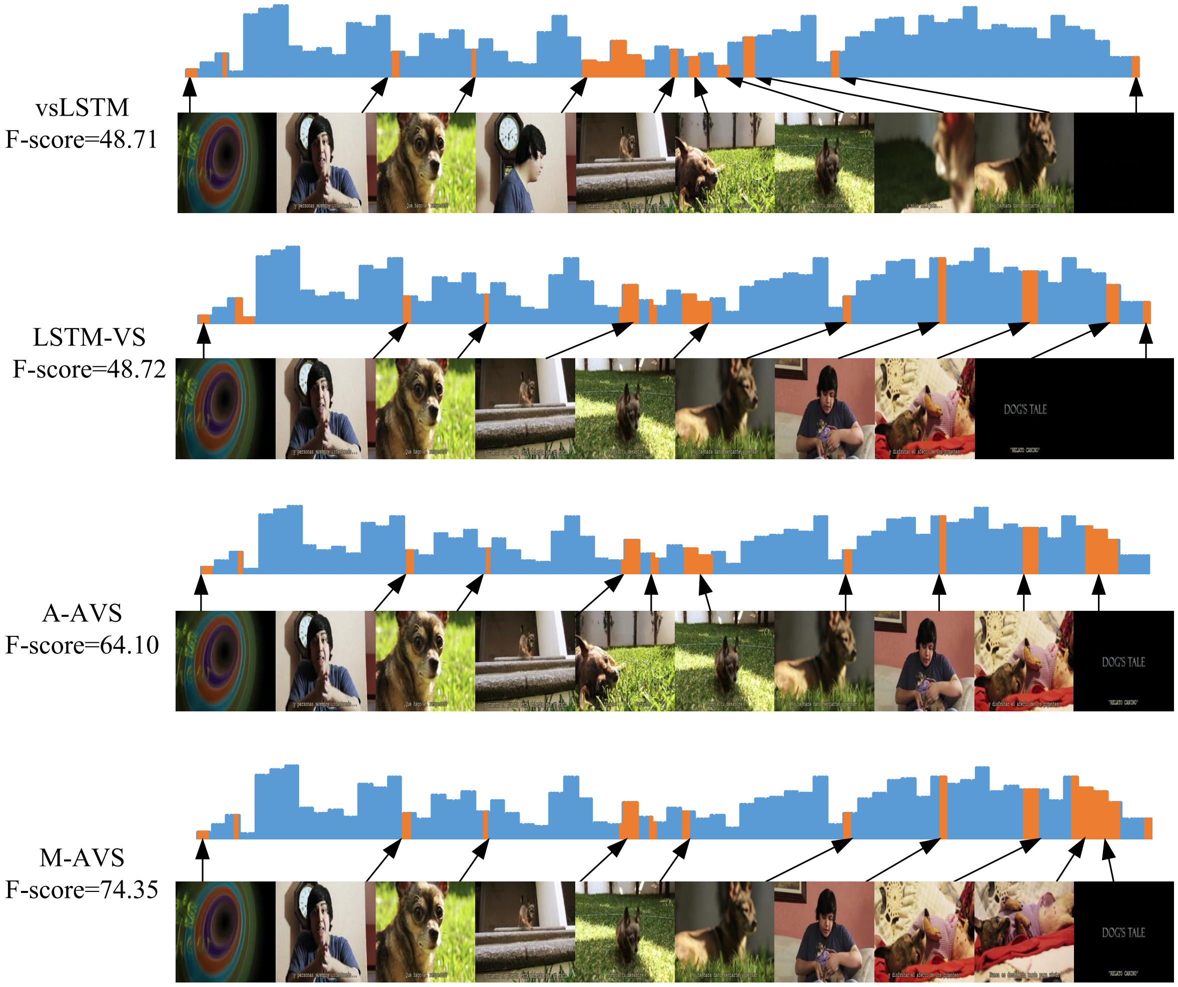


Fig. 7. Exemplar video summaries (orange intervals) from a sample video (the 48th video of TVSum) along with the ground-truth importance scores (blue background).

named A-AVS and M-AVS are developed, respectively. To the best of our knowledge, our work is the first attempt to apply attention mechanism in deep models for video summarization. The proposed models outperform the competing methods on three benchmark datasets by 3%-11%. We also provide the qualitative analysis and parameter sensitive analysis. In addition, the augmentation experiments also verify the effectiveness and superiority of AVS framework when applied augmented data.

In our future work, we will explore more sophisticated attention mechanism in the proposed AVS framework to obtain richer contextual information. Moreover, the existing datasets are not large enough in scale. Thus, the insufficient training data restrict the performance and development of supervised video summarization approaches. To address this problem, we will apply transfer learning [41] and *Generative Adversarial Network* (GAN) [18] techniques to the proposed AVS framework.

References

[1] Ba Tu Truong, and Svetha Venkatesh, “Video abstraction: a systematic review and classification,” *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 3, no. 1, pp. 1–37, 2007.

[2] Arthur G. Money, and Harry Agius, “Video summarisation: a conceptual framework and survey of the state of the art,” *J. Vis. Commun. Image Represent.*, vol. 12, no. 2, pp. 121–143, 2008.

[3] Yufei Ma, Lie Lu, Hongjiang Zhang, and Mingjing Li, “A user attention model for video summarization,” in *Proc. ACM Multimedia*, 2002, pp. 533–542.

[4] Meng Wang, Richang Hong, Guangda Li, Zhengjun Zha, and Shuicheng Yan, “Event driven web video summarization by tag localization and key-shot identification,” *IEEE Trans. Multimedia,* vol. 14, no. 4, pp. 975–985, 2012.

[5] Yunzuo Zhang, Ran Tao, and Yue Wang, “Motion-state-adaptive video summarization via spatiotemporal analysis,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27 , no. 6, pp. 1340–1352, 2017.

[6] Xun Xu, Timothy M. Hospedales, and Shaogang Gong, “Discovery of shared semantic spaces for multiscene video query and summarization,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 6, pp. 1353–1367, 2017.

[7] Richang Hong, Lei Li, Junjie Cai, Dapeng Tao, Meng Wang, and Qi Tian, “Coherent semantic-visual indexing for large-scale image retrieval in the cloud,” *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4128–4138, 2017.

[8] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and

Xinbo Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *Proc. AAAI Conf. Art. Intell.*, 2017, pp. 1618–1625.

[9] Yachuang Feng, Yuan Yuan, and Xiaoqiang Lu, "Learning deep event models for crowd anomaly detection," *Neurocomput.*, vol. 219, pp. 548–556, 2017.

[10] Chong Wah Ngo, Yufei Ma, and Hongjiang Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 296–305, 2005.

[11] Zuzana Cernekova, Ioannis Pitas, and Christophoros Nikou, "Information theory-based shot cut/fade detection and video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 82–90, 2006.

[12] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman, "Discovering important people and objects for egocentric video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1346–1353.

[13] Aditya Khosla, Raffay Hamid, Chih-Jen Lin and Neel Sundaresan, "Large-scale video summarization using web-image priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2698–2705.

[14] Boqing Gong, Weilun Chao, Kristen Grauman, and Fei Sha, "Diverse sequential subset selection for supervised video summarization," in *Advances Neural Inf. Process. Syst.*, 2014, pp. 2069–2077.

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

[16] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc. Van Gool, "Creating summaries from user videos," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 505–520.

[17] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman, " Video summarization with long short-term memory," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 766–782.

[18] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic, "Unsupervised video summarization with adversarial LSTM networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1–10.

[19] Li Xuelong, Bin Zhao, and Xiaoqiang Lu, "A general framework for edited video and raw video summarization," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3652–3664, 2017.

[20] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko, "Sequence to sequence-video to text," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4534–4542.

[21] Dzmitry Bahdanau, Kyungh Yun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," in *Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[22] Minh Thang Luong, Hieu Pham, Christopher D. Manning, "Effective approaches to attention-based neural machine translation," in *Conf. on Empi. Meth. Natural Lan. Proc.*, 2015, pp. 1412–1421.

[23] Huan Yang, Baoyuan Wang, Stephen Lin, David Wipf, Minyi Guo, and Baining Guo, "Unsupervised extraction of video highlights via robust recurrent auto-encoders," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4633–4641.

[24] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid, "Category-specific video summarization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 540–555.

[25] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes, "TVSum: summarizing web videos using titles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5179–5187.

[26] Ejaz, Naveed, Irfan Mehmood, and Sung Wook Baik, "Efficient visual attention based framework for extracting key frames from videos," *Signal Process. Image Commun.*, vol. 28, no. 1, pp. 34–44, 2013.

[27] Sandra Eliza Fontes de Avila, Ana Paula Brando Lopes, Antonio da Luz Jr., and Arnaldo de Albuquerque Arajo, "VSUMM: a mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognit. Lett.*, vol. 32, no. 1, pp. 56–68, 2011.

[28] Sanjay K. Kuanar, Rameswar Panda, and Ananda. S. Chowdhury, "Video key frame extraction through dynamic delaunay clustering with a structural constraint," J. *Vis. Commun. Image Represent.*, vol. 24, no. 7, pp. 1212–1227, 2013.

[29] Shaohui Mei, Genliang Guan, Zhiyong Wang, Mingyi He, and David Dagan Feng, "Video summarization via minimum sparse reconstruction," *Pattern Recognition*, vol. 48, no. 2, pp. 522–533, 2015.

[30] Yang Cong, Junsong Yuan, and Jiebo Luo, "Towards scalable summarization of consumer videos via sparse dictionary selection," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 66–75, 2012.

[31] Zhong Ji, Yaru Ma, Yanwei Pang, and Xuelong Li, "Query-aware sparse coding for multi-video summarization," http://arxiv.org/abs/1707.04021, 2017.

[32] Gunnar A. Sigurdsson, Xinlei Chen, Abhinav Gupta, "Learning visual storylines with skipping recurrent neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 71–88.

[33] Keivin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio, "Show, attend and tell: neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[34] Adway Mitra, Soma Biswas, and Chiranjib Bhattacharyya. "Bayesian modeling of temporal coherence in videos for entity discovery and summarization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 430–443, 2017.

[35] Haykel Boukadida, Sid-Ahmed Berrani, and Patrick Gros, "Automatically creating adaptive video summaries using constraint satisfaction programming: Application to sport content," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 4, pp. 920–934, 2017.

[36] Open Video Project. http://www.open-video.org/.

[37] Liqiang Nie, Richang Hong, Luming Zhang, Yingjie Xia, Dacheng Tao, and Nicu Sebe, "Perceptual attributes optimization for multivideo summarization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2991–3003, 2016.

[38] Alex Graves, and Jrgen Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *Int. Joint Conf. on Neural Net.*, vol. 4, 2005, pp. 2047–2052.

[39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[40] Yanwei Fu, Yanwen Guo, Yanshu Zhu, Feng Liu, Chuanming Song, and Zhihua Zhou, "Multi-View Video Summarization," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 717–729, 2010.

[41] Ke Zhang, Weilun Chao, Fei Sha and Kristen Grauman, "Summary transfer: exemplar-based subset selection for video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1059–1067.