Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Video summarization with a graph convolutional attention network*

Ping LI[†‡1,2], Chao TANG[1], Xianghua XU[1]

[1]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China*

[2]*The State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China*

[†]E-mail: patriclouis.lee@gmail.com

Received Aug. 25, 2020; Revision accepted Jan. 14, 2021; Crosschecked Apr. 1, 2021

Abstract: Video summarization has established itself as a fundamental technique for generating compact and concise video, which alleviates managing and browsing large-scale video data. Existing methods fail to fully consider the local and global relations among frames of video, leading to a deteriorated summarization performance. To address the above problem, we propose a graph convolutional attention network (GCAN) for video summarization. GCAN consists of two parts, embedding learning and context fusion, where embedding learning includes the temporal branch and graph branch. In particular, GCAN uses dilated temporal convolution to model local cues and temporal self-attention to exploit global cues for video frames. It learns graph embedding via a multi-layer graph convolutional network to reveal the intrinsic structure of frame samples. The context fusion part combines the output streams from the temporal branch and graph branch to create the context-aware representation of frames, on which the importance scores are evaluated for selecting representative frames to generate video summary. Experiments are carried out on two benchmark databases, SumMe and TVSum, showing that the proposed GCAN approach enjoys superior performance compared to several state-of-the-art alternatives in three evaluation settings.

Key words: Temporal learning; Self-attention mechanism; Graph convolutional network; Context fusion; Video summarization

https://doi.org/10.1631/FITEE.2000429 **CLC number:** TP391

1 Introduction

The overwhelming popularity of camera-assisted terminal devices has incurred tremendous difficulties in managing and browsing large volumes of diverse videos. As reported in the Cisco Global Networking Trends Report (Cisco, 2020), Internet video will represent 82% of all business Internet traffic, and Internet video surveillance traffic will increase sevenfold by 2022. In this increasingly demanding environment with the sheer scale of data, there is a critical need to interpret the video contents in a compressed and condensed form by eliminating redundant frames, which can be achieved by video summarization techniques (Gong et al., 2014; Basavarajaiah and Sharma, 2019). Tremendous applications including action recognition, security surveillance, film advertisement, and TV broadcasting have motivated the advancement of video summarization for a long time. Basically, the primary goal is to reshape the original video in a compact summary form, which is composed of a sequence of representative and diverse key frames in temporal order. For a long-duration movie, the compact summary tells an interesting storyline that highlights the

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61872122 and 61502131), the Zhejiang Provincial Natural Science Foundation of China (No. LY18F020015), the Open Project Program of the State Key Lab of CAD&CG, China (No. 1802), and the Zhejiang Provincial Key Research and Development Program, China (No. 2020C01067)

ORCID: Ping LI, https://orcid.org/0000-0002-8515-7773

film actions made in the movie.

In Fig. 1, the top row shows the video frames decoded from the factory explosion video from the Tencent Video channel, the middle row indicates whether the current frame is a key frame or not, and the bottom row illustrates the video summary composed of the chosen key frames that record the most important shot revealing the conflagration and dense smoke.

Traditional methods adopt hand-crafted features for video summarization, such as HSV (hue, saturation, value) color feature descriptors (Lei et al., 2014), the texture and color feature (Mahmoud et al., 2013), the scale-invariant feature transform (SIFT) descriptor (Guan et al., 2013), and the SIFT-point distribution histogram (Hannane et al., 2016). In addition, they use different ways to better guide the generation of the video summary, such as the k-means clustering (de Avila et al., 2011), ℓ_0-sparsity constraint (Mei et al., 2015), and locality-constrained linear encoding (Lu et al., 2014). Although these features can capture the local or global relations of video frames in some sense, they are isolated from the generated summary and cannot reflect temporal relations when evaluating the importance of video frames. Recently, deep learning methods have been found to have impressive performance in

Fig. 1 Example of video summarization

many applications including video object segmentation (Chen et al., 2020), crowd counting (Huang et al., 2020), and video summarization (Zhang et al., 2016; Mahasseni et al., 2017; Yuan L et al., 2019). For example, to model the variable-length dependency among frames, video summarization is regarded as sequence-to-sequence learning with long short-term memory (LSTM) units (Zhang et al., 2016; Mahasseni et al., 2017). Similar to the encoder-decoder framework, Yuan L et al. (2019) proposed adversarial LSTM networks to achieve cycle consistency for generating summaries in an unsupervised manner, whereas Ji et al. (2020) used attention-based LSTM networks for the decoder to better select the key video shots. Differently, Zhou et al. (2018) adopted a reinforcement learning framework to obtain a satisfactory summary from the diversity and representativeness perspectives.

However, there are still two drawbacks in the previous approaches: one drawback is that both the local and global temporal relations among video frames are not fully considered, which leads to inferior summarization performance; the other is that they fail to respect the intrinsic structure of frame samples, which is important to encode the semantic relations within key frames. Inspired by recent works (Kipf and Welling, 2017; Li JN et al., 2019), we address the above issues using graph convolutional attention networks (GCANs) for video summarization.

As depicted in Fig. 2, the GCAN framework consists of two parts, i.e., embedding learning and context fusion. Embedding learning aims to learn

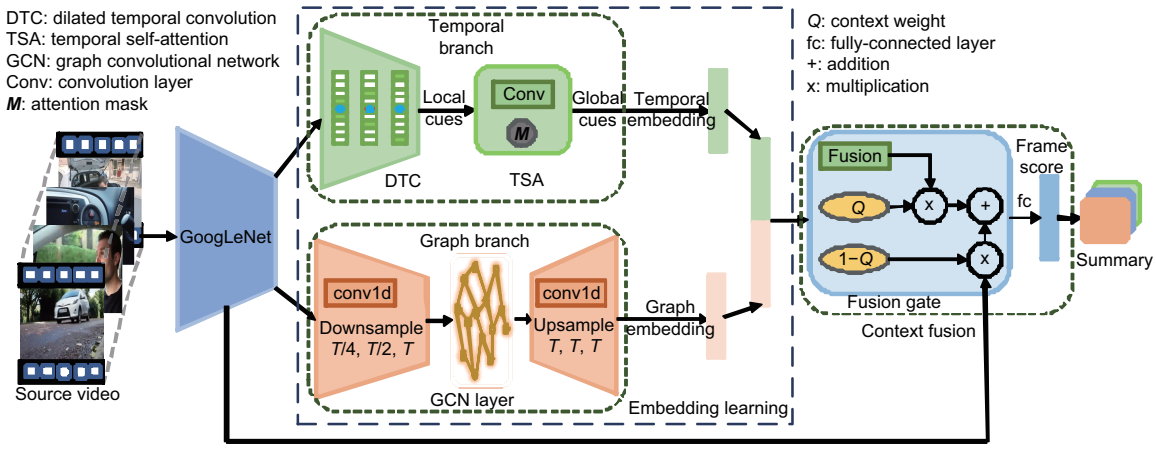

Fig. 2 Framework of the graph convolutional attention network (GCAN)

temporal graph embedding from video, whereas context fusion captures the structured context-aware representation of video. Embedding learning is achieved by going through two parallel branches, including a temporal branch and a graph branch, which respectively generate two kinds of embedding, i.e., temporal embedding and graph embedding. In the temporal branch, GCAN adopts dilated temporal convolution (DTC) to model local cues and temporal self-attention (TSA) to exploit global cues, for video frames. The local temporal cue among adjacent frames can discriminate spatially similar ones, and the global cue is robust to occlusion and noise in long-range video. In the graph branch, GCAN adopts multi-layer graph convolutional network based encoder-decoder to derive graph embedding of frames, which reveals the intrinsically spatial structure of frame samples. The context fusion part combines the output streams from the temporal branch and graph branch to capture context-aware representation with the graph structure of data samples. In this way, the spatial patterns and temporal relations of video frames are well taken into account by GCAN, which is beneficial for selecting frames with more representativeness and less redundancy, resulting in satisfactory video summary for demanding applications.

The main contributions of this work can be highlighted below:

1. Temporal embedding and graph embedding of video frames are learned in a unified framework, i.e., GCAN, to make the frame embedding capture the dynamic nature of video with more discriminative and robust video summarization ability.

2. The fusion gate in GCAN is designed to simultaneously consider the temporal context among adjacent frames and the graph context from the entire video. It paves the way for producing a well-structured context-aware representation for embedding semantic video information, so the importance score of frames with respect to the video summary can be evaluated in a more rational way.

3. Extensive experiments are conducted on two benchmark databases, i.e., SumMe (Gygli et al., 2014) and TVSum (Song et al., 2015), to investigate GCAN summarization performance in both supervised and unsupervised forms. Results validate the effectiveness of GCAN compared to many other approaches.

2 Related works

Video summarization has long been a fundamental problem in various applications, e.g., video search in traffic management, video retrieval in security surveillance, and trailer or ad generation. Its essential goal is to select a collection of representative frames as a compact summary of the original video. Traditionally, clustering (Zhuang et al., 1998; Aner and Kender, 2002; de Avila et al., 2011; Kuanar et al., 2013; Chu et al., 2015) and dictionary learning (Cong et al., 2012, 2017; Elhamifar et al., 2012; Luan et al., 2014; Zhao and Xing, 2014; Mei et al., 2015) have gained popularity in identifying the desired frames, and great deep learning success (Zhang et al., 2016; Mahasseni et al., 2017; Rochan et al., 2018; Zhao et al., 2018; Yuan YT et al., 2019; Li P et al., 2021) is found in video summarization tasks.

In principle, clustering-based methods and dictionary learning methods are both unsupervised approaches, and require that the generated summary be representative, informative, or diverse. The former treats video frames as data points to be grouped into clusters, whose centers indicate the most representative video frames. For example, Zhuang et al. (1998) adopted a clustering algorithm to extract key frames and adapt to visual content of frames at low cost; Kuanar et al. (2013) used dynamic Delaunay graph clustering via an iterative edge-pruning strategy to extract key frames; Chu et al. (2015) described a co-clustering method that groups visually similar pairs of shots into co-clusters, and these shots frequently appear across video frames within the same topic. The latter assumes that key frames or shots are sparsely distributed in video and learns a dictionary that consists of representative frames or shots as basic elements. For example, video summarization was formulated as a minimum sparse reconstruction problem in Mei et al. (2015), where key frames were selected as a sparse dictionary to reconstruct the whole video. Similarly, the representative dictionary was learned with consistent sparsity in Cong et al. (2012). From the nonnegative perspective, Luan et al. (2014) selected key frames by minimizing the nonnegative linear reconstruction function, to process the frames on the fly, and Zhao and Xing (2014) proposed to learn a dictionary using group sparse coding and updating atoms online, which generates a summary composed of video segments that

fail to sparsely reconstruct the learned dictionary. Rather than learning a new dictionary, Cong et al. (2017) used a forward-backward greedy algorithm to directly obtain a compact basis subset from video frames.

Deep learning techniques have largely enhanced the performance of video summarization compared to traditional ones. Usually, a recurrent neural network (RNN) is adopted to encode the temporal relations among video frames. Zhang et al. (2016) modeled the variable-range dependency of frames via directional LSTM and introduced the determinantal point process (DPP) to enhance the diversity of selected frames. To select a sparse subset of frames to represent the video, Mahasseni et al. (2017) designed a generative adversarial framework that includes one summarizer and one discriminator, both of which adopt LSTM as the coding network. Zhao et al. (2018) argued that the video data has a hierarchical structure including shots and corresponding frames, and thus proposed a hierarchical structure-adaptive RNN (HSA-RNN) to generate a summary. Sometimes, the side information associated with video, such as titles, queries, and descriptions, is freely available and is valuable as human-created semantic cues for video summarization. To use the side information, Yuan YT et al. (2019) proposed to select semantically meaningful segments by minimizing the distance between these segments and the side information derived from the latent subspace, which correlates the hidden layers of two auto-encoders. Unlike RNN-based methods, Rochan et al. (2018) proposed a fully convolutional sequence model by adapting popular semantic segmentation networks for video summarization. To extract the most semantically relevant video segments, Wei et al. (2018) developed a semantic attended video summarization network, which employs the frame selector and video descriptor to obtain a subset of shots by minimizing the distance between the generated description of summary video and the human-annotated text of source video. In addition, to reduce the high cost of human annotations, Rochan and Wang (2019) generated a video summary by learning a mapping function from unpaired data with the help of an adversarial objective and a diversity constraint. To track the temporal structure of videos and enforce local diversity, Li YD et al. (2018) built a probabilistic model on sequential DPP to dynamically control the time span of video segments, and this model is trained by a reinforcement learning algorithm. In fact, the video summarization task not only generates a summary from video, but also infers the original content from the summary, which motivated Zhao et al. (2020) to propose a dual learning framework. This framework combines summary generation and video reconstruction as the primal and dual task, where two property models are designed to capture both the spatial and temporal video summary information.

However, none of the existing methods consider video summarization from a graph perspective, treating the key frames or key shots as graph nodes and exploring the relations among these nodes by graph learning. Local and global video frame cues are overlooked for learning their representations in some sense. In this paper, we primarily address the referenced problems.

3 Proposed GCAN method

The entire architecture of the proposed GCAN framework is depicted in Fig. 2. The video summarization is formulated as a context-aware embedding learning problem, in which the sequence of video frames is accepted as the input. It will generate a small subset of key shots to form the video summary that best describes the video content without losing much semantic information. In this section, the embedding learning and context fusion parts are explained.

3.1 Embedding learning

GCAN learns the temporal embedding and graph embedding of video frames in two individual branches that join together as the input of the context fusion part.

Given a video sequence set $\{f_i\}_{i=1}^{T}$ consisting of T frames, where f_i denotes the i^{th} frame, GoogLeNet (Szegedy et al., 2015) is used to extract frame features $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_T]\in\mathbb{R}^{d\times T}$, where $\boldsymbol{x}_i\in\mathbb{R}^d$ denotes the feature of the i^{th} frame with d elements. These frame features are fed into the temporal branch and graph branch separately for learning corresponding temporal embedding $\boldsymbol{B}=[\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\boldsymbol{b}_T]$ and graph embedding $\boldsymbol{G}=[\boldsymbol{g}_1,\boldsymbol{g}_2,\ldots,\boldsymbol{g}_T]$ (here, b_i $(i=1,2,...,T)$ denotes the temporal embedding vector of the i^{th} frame and g_i denotes the graph embedding vector of of the i^{th} frame).

3.1.1 Temporal branch

The temporal branch includes DTC and TSA modules to exploit local cues and global cues, respectively, for discriminating video frames with different semantics. DTC is beneficial for modeling the spatial context of images (Yu and Koltun, 2016), and GCAN adopts dilated temporal pyramid (Li JN et al., 2019) convolutions with multi-scale dilation rates to improve the ability to modele local temporal cues of video frames. Self-attention also exhibits excellent power in spatial context modeling, and GCAN uses self-attention (Li JN et al., 2019) to encode the global temporal relations across all frames.

To learn temporal embedding of frames, several dilated convolutional kernels $\boldsymbol{W}^r \in \mathbb{R}^{d\times w}$ with temporal width w and dilation rate r are used to capture the spatial context of frames, which output the local embedding set $\{\boldsymbol{x}_i^r\}_{i=1}^T$ formulated as

$$\boldsymbol{x}_i^r = \sum_{j=1}^{w}(\boldsymbol{x}_{i+rj} \odot \boldsymbol{W}_j^r), \tag{1}$$

where $\boldsymbol{x}_i^r \in \mathbb{R}^d, \boldsymbol{W}_j^r \in \mathbb{R}^d$, and the dilation rate r specifies the temporal stride for frame sampling, which indicates the temporal scales for DTC that enlarges the receptive field of neurons without reducing resolution. Here, r is set to 2 and w is set to 3. The symbol "$\odot$" means the element-wise multiplication. The dilated temporal pyramid convolution is composed of N parallel dilated convolution operators with an ascending dilation rate sequence to consider ranges of different lengths. The n^{th} dilation rate $r_n = 2^{n-1}$ is used to increase the receptive field along the temporal dimension. The outputs of those parallel dilated convolutions are concatenated to yield the enhanced local temporal feature of the i^{th} frame, i.e., $\boldsymbol{c}_i = [\boldsymbol{x}_i^{r_1}, \boldsymbol{x}_i^{r_1}, \ldots, \boldsymbol{x}_i^{r_N}] \in \mathbb{R}^{Nd}$, resulting in the local temporal embedding $\boldsymbol{C} \in \mathbb{R}^{Nd\times T}$.

To capture the global relations among frames, the self-attention module immediately follows the DTC module. Its input is the local temporal embedding $\boldsymbol{C} \in \mathbb{R}^{Nd\times T}$, and the output is the temporal embedding $\boldsymbol{B} \in \mathbb{R}^{Nd\times T}$. The TSA module is introduced to compute the attention mask $\boldsymbol{M} \in \mathbb{R}^{T\times T}$, which is obtained by multiplying two feature maps, i.e., $\boldsymbol{M} = \boldsymbol{M}_1 \times \boldsymbol{M}_2$, where $\boldsymbol{M}_1, \boldsymbol{M}_2 \in \mathbb{R}^{\frac{Nd}{2}\times T}$ and constant 2 is used to govern the size of module parameters. The two feature maps are generated by passing two convolution layers, batch normalization and ReLU units. The mask is employed to incorporate video frame global temporal cues in the previous local temporal embedding $\boldsymbol{C}$, which is updated as $\boldsymbol{C}' \in \mathbb{R}^{\frac{Nd}{2}\times T}$ by going through a convolution layer. The updated embedding $\boldsymbol{C}'$ is multiplied by mask $\boldsymbol{M}$, and a convolution layer is added to resize the matrix to $Nd \times T$. Finally, the temporal embedding matrix $\boldsymbol{B}$ is calculated by

$$\boldsymbol{B} = \text{conv1d}(\boldsymbol{C}' \cdot \boldsymbol{M}) + \boldsymbol{C}, \tag{2}$$

where conv1d$(\cdot)$ denotes the one-dimensional convolution with kernel size 1, stride 1, input channel size $d/2$, and output channel size d, such that the size of the learned embedding can be recovered to $Nd \times T$, and the local temporal embedding $\boldsymbol{C}$ is added through a residual connection.

3.1.2 Graph branch

The graph branch consists of an encoder, a decoder, and several graph convolutional layers. The encoder is designed to capture local semantics of different temporal scales for the input frame features by downsampling, which reduces the size of feature maps in different scales. Here, for simplicity, we take three downsampling operations for example; two one-dimensional (1D) convolution kernels $\boldsymbol{W}_1, \boldsymbol{W}_2$ with kernel size 3 and stride 2 are used to reduce the sizes of feature maps $\boldsymbol{X}$ and $\boldsymbol{X}_1$ to $T/2$ and $T/4$, respectively:

$$\boldsymbol{X}_1 = \boldsymbol{X}^{\mathrm{T}} * \boldsymbol{W}_1 \in \mathbb{R}^{\frac{T}{2}\times d}, \tag{3}$$

$$\boldsymbol{X}_2 = \boldsymbol{X}_1 * \boldsymbol{W}_2 \in \mathbb{R}^{\frac{T}{4}\times d}, \tag{4}$$

where the symbol "$(\cdot)^{\mathrm{T}}$" denotes the transpose operation, "$*$" denotes the convolution operation, and the 1D convolution kernel slides along the temporal dimension. The feature map $\boldsymbol{X}_1$ contains the local temporal information of three adjacent frames, while $\boldsymbol{X}_2$ contains that of five adjacent frames. Then three 1D convolution kernels Ψ, Ψ_1, and Ψ_2 with kernel size 1 and stride 1 are carried out on the original frame feature $\boldsymbol{X}$ and two feature maps $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$:

$$\boldsymbol{X}' = \boldsymbol{X} * \Psi \in \mathbb{R}^{T\times d}, \tag{5}$$

$$\boldsymbol{X}_1' = \boldsymbol{X}_1 * \Psi_1 \in \mathbb{R}^{\frac{T}{2}\times d}, \tag{6}$$

$$\boldsymbol{X}_2' = \boldsymbol{X}_2 * \Psi_2 \in \mathbb{R}^{\frac{T}{4}\times d}. \tag{7}$$

The updated feature maps $\boldsymbol{X}'$, $\boldsymbol{X}_1'$, and $\boldsymbol{X}_2'$ are concatenated into a unified feature map with

multi-scale local temporal semantics, i.e., $\tilde{\boldsymbol{X}} = [\boldsymbol{X}', \boldsymbol{X}'_1, \boldsymbol{X}'_2] \in \mathbb{R}^{c\times d}$ $(c = T + T/2 + T/4)$, which is fed into the graph convolutional network (GCN) layer.

The GCN model $f(\tilde{\boldsymbol{X}}, \boldsymbol{A})$ generates the graph representation of $\tilde{\boldsymbol{X}}$ with matrix $\boldsymbol{A} \in \mathbb{R}^{c\times c}$ encoding the pairwise relationship among video frames. The proposed GCAN method adopts the multi-layer GCN following the layer-wise propagation rule in L $(L = 2)$ hidden layers (Kipf and Welling, 2017):

$$\boldsymbol{H}^{(l+1)} = \sigma\left(\boldsymbol{D}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right), \tag{8}$$

where $\boldsymbol{H}^{(l)} \in \mathbb{R}^{c\times d_l}$ denotes the output of activations in the l^{th} layer $(l = 0, 1, \ldots, L-1$ and $\boldsymbol{H}^{(0)} = \tilde{\boldsymbol{X}})$, and $\sigma(\cdot)$ is an activation function, e.g., $\text{ReLu}(\cdot) = \max(0, \cdot)$. The matrix $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix of the graph with added self-connections, $\boldsymbol{I} \in \mathbb{R}^{c\times c}$ is the identity matrix, the diagonal elements of the diagonal matrix $\boldsymbol{D}$ are computed as $D_{ii} = \sum_j \tilde{A}_{ij}$, and the weight matrix $\boldsymbol{W}^l \in \mathbb{R}^{d_l\times d_{l+1}}$ is layer-specific trainable. For video summarization, the dimension of the weight matrix in each hidden layer is set to d. The adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{c\times c}$ is obtained by $\text{ReLU}(\cdot)$ activation using two 1D convolution kernels Φ_1 and Φ_2 with kernel size 1 and stride 1 on feature map $\tilde{\boldsymbol{X}}$:

$$\boldsymbol{A} = \sigma([\tilde{\boldsymbol{X}} * \Phi_1] \times [\tilde{\boldsymbol{X}} * \Phi_2]^{\mathrm{T}}). \tag{9}$$

Since the sum of each column of the learned graph representation $\tilde{\boldsymbol{A}}$ is 1 and the entries in $\tilde{\boldsymbol{A}}$ are nonnegative, Eq. (8) can be compacted as

$$\boldsymbol{H}^{(l+1)} = \sigma\left(\tilde{\boldsymbol{A}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right). \tag{10}$$

The output of the multi-layer GCN is $\boldsymbol{G}_{\text{in}} = [\boldsymbol{H}^{(l)}]' \in \mathbb{R}^{c\times d}$, which is fed into the decoder that contains three upsampling operations to capture the multi-scale temporal semantics (Fig. 3). Specifically, linear interpolations are performed on different parts of matrix $\boldsymbol{G}_{\text{in}}$, which is applied with three temporal scales, i.e., T, $T/2$, and $T/4$, leading to three $d \times T$ upsampled matrices. Then, the average pooling is applied to these matrices, resulting in the graph embedding matrix $\boldsymbol{G} \in \mathbb{R}^{d\times T}$.

3.2 Context fusion

To capture the context-aware representation with graph structure, a fusion gate is designed in GCAN to integrate the temporal embedding $\boldsymbol{B} \in \mathbb{R}^{Nd\times T}$ with the graph embedding $\boldsymbol{G} \in \mathbb{R}^{d\times T}$. In particular, two kinds of information are concatenated into a unified representation $\boldsymbol{Z} = [\boldsymbol{B}, \boldsymbol{G}] \in \mathbb{R}^{(N+1)d\times T}$. The context fusion module includes a linear layer with an activation function, a linear layer with a sigmoid function, a fusion gate (Shen et al., 2018), and a fully-connected layer. The mathematical formulations of two linear layers are as follows:

$$\boldsymbol{Z}' = \text{ReLu}(\boldsymbol{W}_{\text{z}}\boldsymbol{Z}) \in \mathbb{R}^{d\times T}, \tag{11}$$

$$\boldsymbol{Q} = \text{sigmoid}(\boldsymbol{W}_{\text{g}}\boldsymbol{Z}) \in \mathbb{R}^{d\times T}, \tag{12}$$

where $\boldsymbol{W}_{\text{z}} \in \mathbb{R}^{d\times(N+1)d}$ and $\boldsymbol{W}_{\text{g}} \in \mathbb{R}^{d\times(N+1)d}$ are the weight matrices to be learned, $\boldsymbol{Z}'$ and $\boldsymbol{Q}$ are the outputs of the two linear layers, and $\boldsymbol{Q}$ can be treated as the context weight matrix for the learned representation $\boldsymbol{Z}'$. To preserve the semantics of the original frames, the frame feature matrix $\boldsymbol{X}^{d\times T}$ and the graph structured temporal embedding matrix $\boldsymbol{Z}'$ are fused by a fusion gate, leading to the context-aware representation $\boldsymbol{Z}_f \in \mathbb{R}^{d\times T}$:

$$\boldsymbol{Z}_f = \boldsymbol{Z}' \odot \boldsymbol{Q} + \boldsymbol{X} \odot (1 - \boldsymbol{Q}). \tag{13}$$

The context-aware representation $\boldsymbol{Z}_f$ is fed into one fully-connected layer and outputs the nonnegative importance score of each frame, i.e., $\{s_i\}_{i=1}^{T}$.

Once the importance scores of all frames are available, kernel temporal segmentation (KTS) (Potapov et al., 2014) is used to select the key shots for generating video summary. KTS segments the visually similar frames in shots and calculates the shot-level importance scores.

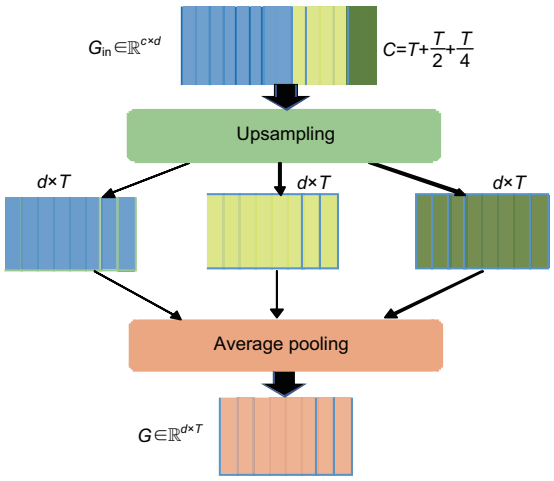

Fig. 3 Decoder of the graph branch

3.3 Loss function

GCAN adopts the binary cross-entropy loss to optimize the parameters in the model, i.e.,

$$\mathcal{L}_{\text{ce}}(s,y)=-\frac{1}{T}\sum_{i=1}^{T}y_i\log s_i+(1-y_i)\log(1-s_i), \tag{14}$$

where $y_i\in\mathbb{R}$ is the user-generated annotation of the i^{th} frame. So, GCAN is essentially a supervised video summarization method.

However, the human labeling cost is usually high, and it is desirable to develop an unsupervised approach to solve this problem. Thus, the unsupervised version $\text{GCAN}_{\text{unsup}}$ adopts the following sparsity loss:

$$\mathcal{L}_{\text{sparsity}}(s,\epsilon)=\left\|\frac{1}{T}\sum_{i=1}^{T}s_i-\epsilon\right\|_2, \tag{15}$$

where the constant $\epsilon>0$ is used to specify the percentage of frames expected to be selected in the summary, and $\|\cdot\|_2$ stands for the 2-norm operation. Here, the constant ϵ is fixed to 0.3 as suggested in Mahasseni et al. (2017), which regards the above formulation as the summary-length regularization term.

4 Experiments

This section presents the experimental results of the proposed GCAN approach in three evaluation settings, as well as the ablation study of the temporal branch and the graph branch of GCAN. The descriptions of data sets, evaluation metric, evaluation setting, and implementation details are provided.

4.1 Data sets

The performance of the proposed GCAN is examined on two benchmark data sets, i.e., SumMe (Gygli et al., 2014) and TVSum (Song et al., 2015). SumMe contains 25 user videos with a length of 1–6.5 min that records various events such as sports and holidays, where the scene changes quickly or slowly. TVSum has 50 edited videos in 10 categories with a length of 1.5–11 min. Both data sets have user annotations that indicate the frame-level importance scores of each video. Following Zhang et al. (2016) and Jung et al. (2019), two additional data sets OVP (de Avila et al., 2011) with 50 videos and YouTube (de Avila et al., 2011) with 39 videos are used to evaluate the augmented setting and transfer setting.

4.2 Evaluation metric

The F-measure score (Zhang et al., 2016; Jung et al., 2019; Ji et al., 2020; Zhao et al., 2020) is widely used to evaluate the summarization performance. For all data sets, user annotations are changed from frame-level scores to key shot scores using KTS (Potapov et al., 2014), and the key shots are selected to generate a summary that is less than 15% of the original video duration. The frames with the highest score in each key shot are selected to form the final video summary.

Because F-measure is actually the harmonic mean of precision and recall, it is necessary to compute the precision and recall first. Given the generated summary S_{gen} and human-created summary S_{human}, it defines precision $=\frac{|S_{\text{gen}}\cup S_{\text{human}}|}{|S_{\text{human}}|}$ and recall $=\frac{|S_{\text{gen}}\cup S_{\text{human}}|}{|S_{\text{gen}}|}$, where the numerator denotes the temporal overlaps between each pair of S_{gen} and S_{human} whose duration is indicated by the denominator. Thus, the F-measure score is calculated as

$$\text{F-score}=\frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}\times 100\%. \tag{16}$$

4.3 Evaluation setting

Following the protocol in Zhang et al. (2016), the proposed approach is evaluated in three settings, i.e., canonical (C), augmented (A), and transfer (T) settings, as shown in Table 1. The training set takes up 80% of the videos and the test set takes up 20%. Five-fold cross validation is conducted to obtain the final F-score averaged over the five tests.

4.4 Implementation details

For fair comparison, the frames are downsampled at 2 frames per second as in Zhang et al. (2016) and Ji et al. (2020), and the feature is extracted with 1024 dimensions through the pool5 layer of GoogLeNet (Szegedy et al., 2015) trained on ImageNet. The model adopts the Adam strategy for optimization. The training terminates after 100 epochs and stops if five consecutive epochs have decreasing F-score values. The weights of the GCAN model are randomly initialized and the parameters are listed in Tables 2 and 3 for supervised and unsupervised modes, respectively. All tests

are carried out on a machine with Intel i7-6950X CPU@3.00 GHz and an NVIDIA TITAN Xp GPU card using the PyTorch platform.

4.5 Results and analysis

The proposed GCAN approach can be trained with or without user annotations, leading to two modes, i.e., supervised mode GCAN_{sup} and unsupervised mode $\text{GCAN}_{\text{unsup}}$. Both models are compared to various state-of-the-art alternatives, whose results are recorded from the original papers. The results of GCAN_{sup} and other supervised methods are shown in Table 4, and the results of $\text{GCAN}_{\text{unsup}}$ and other unsupervised methods are shown in Table 5.

From Table 4, it can be observed that the proposed GCAN_{sup} enjoys more satisfactory performance compared to other supervised approaches; e.g., GCAN_{sup} outperforms the state-of-the-art alternative $\text{CSNet}_{\text{sup}}$ by a large margin of 4.4 in the canonical setting on SumMe. This is because with the human labeling information, GCAN can better capture the discriminant context-aware semantic representation of video frames via the embedding learning and context fusion components. The methods vsLSTM (Zhang et al., 2016) and dppLSTM (Zhang et al., 2016) obtain poor video summarization performance, because they adopt LSTM to model the variable-range temporal dependency among video frames and cannot effectively respect the sequential structure of long-range videos due to the limitedness of LSTM. GCAN has used TSA to alleviate this problem by modeling global semantic frame cues. SUM-GAN (Mahasseni et al., 2017) is an essentially unsupervised generative adversarial

Table 1 Evaluation settings

Data set	Setting	Training	Testing
SumMe	C	80% SumMe	20% SumMe
	A	OVP+YouTube+TVSum+80% SumMe	20% SumMe
	T	OVP+YouTube+TVSum	SumMe
TVSum	C	80% TVSum	20% TVSum
	A	OVP+YouTube+SumMe+80% TVSum	20% TVSum
	T	OVP+YouTube+SumMe	TVSum

C: canonical; A: augmented; T: transfer

Table 2 Parameters of GCAN_{sup} in different settings

Data set	Setting	Learning rate	Weight decay	n_{conv}	n_{scale}	$\text{Dropout}_{\text{temp}}$	$\text{Dropout}_{\text{graph}}$
SumMe	C	1×10^{-4}	1×10^{-5}	4	2	0.7	0.5
	A	5×10^{-3}	1×10^{-5}	3	2	0.7	0.5
	T	1×10^{-4}	1×10^{-5}	2	3	0.5	0.7
TVSum	C	5×10^{-3}	5×10^{-3}	3	3	0.6	0.7
	A	5×10^{-3}	5×10^{-3}	3	2	0.5	0.5
	T	5×10^{-3}	1×10^{-5}	2	2	0.7	0.7

n_{conv} is the number of parallel dilated convolutions in the temporal branch, and n_{scale} is the number of downsampling operations in the graph branch. The subscript temp means the temporal branch, and graph means the graph branch. C: canonical; A: augmented; T: transfer

Table 3 Parameters of $\text{GCAN}_{\text{unsup}}$ in different settings

Data set	Setting	Learning rate	Weight decay	n_{conv}	n_{scale}	$\text{Dropout}_{\text{temp}}$	$\text{Dropout}_{\text{graph}}$
SumMe	C	5×10^{-4}	1×10^{-5}	3	3	0.5	0.6
	A	5×10^{-4}	1×10^{-5}	3	3	0.5	0.6
	T	5×10^{-4}	1×10^{-5}	3	3	0.5	0.6
TVSum	C	5×10^{-4}	1×10^{-5}	4	3	0.6	0.7
	A	5×10^{-4}	1×10^{-5}	4	4	0.4	0.5
	T	5×10^{-4}	1×10^{-5}	4	2	0.4	0.6

n_{conv} is the number of parallel dilated convolutions in the temporal branch, and n_{scale} is the number of downsampling operations in the graph branch. The subscript temp means the temporal branch, and graph means the graph branch. C: canonical; A: augmented; T: transfer

Table 4 Performance comparison with state-of-the-art supervised methods

Method	Reference	F-score					
		SumMe			TVSum		
		C	A	T	C	A	T
vsLSTM	Zhang et al. (2016)	37.6	41.6	40.7	54.2	57.9	56.9
dppLSTM	Zhang et al. (2016)	38.6	42.9	41.8	54.7	59.6	58.7
SUM-GAN$_{sup}$	Mahasseni et al. (2017)	41.7	43.6		56.3	61.2	
DR-DSN$_{sup}$	Zhou et al. (2018)	42.1	43.9	42.6	58.1	59.8	**58.9**
HSA-RNN	Zhao et al. (2018)		44.1			59.8	
DySeqDPP	Li YD et al. (2018)	44.3			58.4		
SASUM$_{sup}$	Wei et al. (2018)	45.3			58.2		
SUM-FCN	Rochan et al. (2018)	47.5	51.1	44.1	56.8	59.2	58.2
UnpairedVSN$_{psup}$	Rochan and Wang (2019)	48.0			56.1		
CSNet$_{sup}$	Jung et al. (2019)	48.6	48.7	44.1	58.5	57.1	57.4
A-AVS	Ji et al. (2020)	43.9	44.6		59.4	60.8	
M-AVS	Ji et al. (2020)	44.4	46.1		**61.0**	**61.8**	
PCDL$_{sup}$	Zhao et al. (2020)	43.7	44.1		59.2	61.3	
GCAN$_{sup}$	This paper	**53.0**	**54.2**	**46.8**	60.7	61.1	58.7

The best records in each column are highlighted in bold, and the second-best ones are underlined explicitly. C: canonical; A: augmented; T: transfer

Table 5 Performance comparison with state-of-the-art unsupervised methods

Method	Reference	F-score					
		SumMe			TVSum		
		C	A	T	C	A	T
SUM-GAN$_{rep}$	Mahasseni et al. (2017)	38.5	42.5		51.9	59.3	
SUM-GAN$_{dpp}$	Mahasseni et al. (2017)	39.1	43.4		51.8	**59.5**	
DR-DSN	Zhou et al. (2018)	41.4	42.8	42.4	57.6	58.4	57.8
SASUM	Wei et al. (2018)	40.6			53.9		
Cycle-SUM	Yuan L et al. (2019)	41.9			57.6		
UnpairedVSN	Rochan and Wang (2019)	47.5		41.6	55.6		55.7
CSNet	Jung et al. (2019)	51.3	52.1	**45.1**	58.8	59.0	59.2
PCDL	Zhao et al. (2020)	42.7			58.4		
GCAN$_{unsup}$	This paper	**59.2**	**58.0**	44.6	**59.0**	58.8	**59.7**

The best records in each column are highlighted in bold, and the second-best ones are underlined explicitly. C: canonical; A: augmented; T: transfer

framework composed of the summarizer and the discriminator, both of which also use LSTM because the module inherits the drawback of failing to model the long-term dependency among frames. DR-DSN (Zhou et al., 2018) is an end-to-end reinforcement learning framework that uses a reward function to jointly account for video summary diversity and representativeness, which does improve the performance over SUM-GAN. However, DR-DSN is difficult and expensive to train with a policy gradient. HSA-RNN (Zhao et al., 2018) uses bidirectional LSTM to capture both the forward and backward information in the structure-adaptive summarization framework, but the local-range relations of LSTM limit its ability to cover the global relations of the long-duration video. DySeqDPP (Li YD et al., 2018) improves SeqDPP (Gong et al., 2014) by a latent variable to dynamically control the time span of a video segment where it defines the local diversity using a conditional DPP, and a reinforcement learning algorithm is developed for DySeqDPP, which is also not easy to train. SASUM (Wei et al., 2018) requires text descriptions of video, which are usually unavailable, or adopts a video descriptor to generate descriptions, which is costly and not that accurate for summarization tasks. SUM-FCN (Rochan et al., 2018) further improves the performance (the F-score is 51.1 in an augmented setting on SumMe) by introducing fully convolutional networks originally applied in semantic segmentation, but does not consider the

local and global temporal cues of video frames, which play an important role in generating a high-quality summary. UnpairedVSN (Rochan and Wang, 2019) borrows the adversarial loss to discriminate the distribution of the generated summary video from that of the original video. This GAN-based idea is also used by SUM-GAN (Mahasseni et al., 2017), whose training is a bit unstable. CSNet (Jung et al., 2019) provides a two-stream network to use both local and global frame feature information, but it fails to consider the graph structure of the data. AVS (Ji et al., 2020) uses additive (A-AVS) or multiplicative (M-AVS) functions, employs bidirectional LSTM in the attentive encoder-decoder framework, and achieves satisfactory performance on TVSum, but it neglects the intrinsic structure of frame samples that is critical for capturing context-aware semantics in the video. PCDL (Zhao et al., 2020) captures both spatial and temporal information via the dual learning framework, which also uses bidirectional LSTM, but it cannot encode the long-range relations among video frames.

From Table 5, the proposed $\text{GCAN}_{\text{unsup}}$ also shows superiority in comparison with other unsupervised alternatives. This consolidates the effectiveness of learning both temporal embedding and graph embedding of video frames. Except Cycle-SUM (Yuan L et al., 2019), other compared methods have been discussed previously, and the difference lies in whether they use human-created annotations or not. Cycle-SUM provides only an unsupervised version, and adopts the GAN framework using bidirectional LSTM, which is unable to sufficiently consider the global relations among video frames.

4.6 Ablation study

To investigate the individual components of the proposed GCAN_{sup} approach, the ablation analysis is shown in Table 6. Here, the state-of-the-art method $\text{CSNet}_{\text{sup}}$ (Jung et al., 2019) is listed as the baseline. $\text{GCAN}_{\text{temp}}$ is one variant of GCAN to verify the effectiveness of the temporal learning branch, while $\text{GCAN}_{\text{graph}}$ is another variant for examining the performance of the graph learning branch.

From these records, it can be found that both the temporal branch and graph branch achieve higher F-scores in all three evaluation settings on SumMe and TVSum compared to the baselines (except in the canonical setting on TVSum of the temporal branch). This demonstrates that the derived temporal embedding can accurately model local and global cues from video frames, whose intrinsic structure is encoded by the multi-layer graph convolutional network that captures the global relations of samples. The context fusion part is beneficial for revealing the inherent relations among frames in terms of temporal learning and graph learning viewpoints, which leads to satisfactory video summaries with more representativeness and less redundancy. The corresponding unsupervised version $\text{GCAN}_{\text{unsup}}$ has similar behaviors.

Table 6 Performance comparison of the individual components of GCAN_{sup}

Parameter	F-score					
	SumMe			TVSum		
	C	A	T	C	A	T
$\text{CSNet}_{\text{sup}}$	48.6	48.7	44.1	58.5	57.1	57.4
GCAN_{sup}	53.0	54.2	46.8	60.7	61.1	58.7
$\text{GCAN}_{\text{temp}}$	51.1	51.3	46.1	58.2	58.8	58.1
$\text{GCAN}_{\text{graph}}$	51.5	50.3	44.6	59.3	59.5	57.9

C: canonical; A: augmented; T: transfer

5 Conclusions

In this study, we propose a video summarization approach based on embedding learning and context fusion named the graph convolutional attention network (GCAN). Unlike previous methods that either fail to consider the long-term dependency among frames or neglect the intrinsic structure of the frame samples, GCAN addresses problems in a unified framework that consists of temporal graph embedding learning and context fusion, to capture a graph structured context-aware representation of frames. This allows GCAN to identify key frames with high importance scores, according to which the final video summary is shaped. A number of experiments on benchmark databases have validated the superior performance of GCAN against state-of-the-art approaches in the three evaluation settings. In the future, it will be interesting to explore the use of partial human annotations or progressively select partial frames for labeling, to make video summary generation more economically affordable in demanding applications.

Contributors

Ping LI designed the research. Ping LI and Chao TANG processed the data and drafted the manuscript. Xianghua XU helped organize the manuscript. Ping LI revised and finalized the paper.

Compliance with ethics guidelines

Ping LI, Chao TANG, and Xianghua XU declare that they have no conflict of interest.

References

Aner A, Kender JR, 2002. Video summaries through mosaic-based shot and scene clustering. Proc 7th European Conf on Computer Vision, p.388-402. https://doi.org/10.1007/3-540-47979-1_26

Basavarajaiah M, Sharma P, 2019. Survey of compressed domain video summarization techniques. *ACM Comput Surv*, 52(6):116. https://doi.org/10.1145/3355398

Chen YW, Tsai YH, Lin YY, et al., 2020. VOSTR: video object segmentation via transferable representations. *Int J Comput Vis*, 128(4):931-949. https://doi.org/10.1007/s11263-019-01224-x

Chu WS, Song YL, Jaimes A, 2015. Video co-summarization: video summarization by visual co-occurrence. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.3584-3592. https://doi.org/10.1109/CVPR.2015.7298981

Cisco, 2020. Cisco Global Networking Trends Report. https://www.cisco.com/c/m/en_us/solutions/enterprise-networks/networking-report.html

Cong Y, Yuan JS, Luo JB, 2012. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE Trans Multim*, 14(1):66-75. https://doi.org/10.1109/TMM.2011.2166951

Cong Y, Liu J, Sun G, et al., 2017. Adaptive greedy dictionary selection for web media summarization. *IEEE Trans Image Process*, 26(1):185-195. https://doi.org/10.1109/TIP.2016.2619260

de Avila SEF, Lopes APB, da Luz AJr, et al., 2011. VSUMM: a mechanism designed to produce static video summaries and a novel evaluation method. *Patt Recogn Lett*, 32(1):56-68. https://doi.org/10.1016/j.patrec.2010.08.004

Elhamifar E, Sapiro G, Vidal R, 2012. See all by looking at a few: sparse modeling for finding representative objects. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1600-1607. https://doi.org/10.1109/CVPR.2012.6247852

Gong BQ, Chao WL, Grauman K, et al., 2014. Diverse sequential subset selection for supervised video summarization. Proc 27th Int Conf on Neural Information Processing Systems, p.2069-2077.

Guan GL, Wang ZY, Lu SY, et al., 2013. Keypoint-based keyframe selection. *IEEE Trans Circ Syst Video Technol*, 23(4):729-734. https://doi.org/10.1109/TCSVT.2012.2214871

Gygli M, Grabner H, Riemenschneider H, et al., 2014. Creating summaries from user videos. Proc 13th European Conf on Computer Vision, p.505-520. https://doi.org/10.1007/978-3-319-10584-0_33

Hannane R, Elboushaki A, Afdel K, et al., 2016. An efficient method for video shot boundary detection and keyframe extraction using SIFT-point distribution histogram. *Int J Multim Inform Retr*, 5(2):89-104. https://doi.org/10.1007/s13735-016-0095-6

Huang JH, Di XG, Wu JD, et al., 2020. A novel convolutional neural network method for crowd counting. *Front Inform Technol Electron Eng*, 21(8):1150-1160. https://doi.org/10.1631/FITEE.1900282

Ji Z, Xiong KL, Pang YW, et al., 2020. Video summarization with attention-based encoder-decoder networks. *IEEE Trans Circ Syst Video Technol*, 30(6):1709-1717. https://doi.org/10.1109/TCSVT.2019.2904996

Jung Y, Cho D, Kim D, et al., 2019. Discriminative feature learning for unsupervised video summarization. Proc AAAI Conf on Artificial Intelligence, p.8537-8544. https://doi.org/10.1609/aaai.v33i01.33018537

Kipf TN, Welling M, 2017. Semi-supervised classification with graph convolutional networks. Int Conf on Learning Representations, p.1-14.

Kuanar SK, Panda R, Chowdhury AS, 2013. Video key frame extraction through dynamic Delaunay clustering with a structural constraint. *J Vis Commun Image Represent*, 24(7):1212-1227. https://doi.org/10.1016/j.jvcir.2013.08.003

Lei SS, Xie G, Yan GW, 2014. A novel key-frame extraction approach for both video summary and video index. *Sci World J*, 2014:695168. https://doi.org/10.1155/2014/695168

Li JN, Zhang SL, Wang JD, et al., 2019. Global-local temporal representations for video person re-identification. Proc IEEE/CVF Int Conf on Computer Vision, p.3957-3966. https://doi.org/10.1109/ICCV.2019.00406

Li P, Ye QH, Zhang LM, et al., 2021. Exploring global diverse attention via pairwise temporal relation for video summarization. *Patt Recogn*, 111:107677. https://doi.org/10.1016/j.patcog.2020.107677

Li YD, Wang LQ, Yang TB, et al., 2018. How local is the local diversity? Reinforcing sequential determinantal point processes with dynamic ground sets for supervised video summarization. Proc 15th European Conf on Computer Vision, p.156-174. https://doi.org/10.1007/978-3-030-01237-3_10

Lu SY, Wang ZY, Mei T, et al., 2014. A bag-of-importance model with locality-constrained coding based feature learning for video summarization. *IEEE Trans Multim*, 16(6):1497-1509. https://doi.org/10.1109/TMM.2014.2319778

Luan Q, Song ML, Liau CY, et al., 2014. Video summarization based on nonnegative linear reconstruction. IEEE Int Conf on Multimedia and Expo, p.1-6. https://doi.org/10.1109/ICME.2014.6890332

Mahasseni B, Lam M, Todorovic S, 2017. Unsupervised video summarization with adversarial LSTM networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.2982-2991. https://doi.org/10.1109/CVPR.2017.318

Mahmoud KM, Ghanem NM, Ismail MA, 2013. VGRAPH: an effective approach for generating static video summaries. Proc IEEE Int Conf on Computer Vision Workshops, p.811-818. https://doi.org/10.1109/ICCVW.2013.111

Mei SH, Guan GL, Wang ZY, et al., 2015. Video summarization via minimum sparse reconstruction. *Patt Recogn*, 48(2):522-533.
https://doi.org/10.1016/j.patcog.2014.08.002

Potapov D, Douze M, Harchaoui Z, et al., 2014. Category-specific video summarization. Proc 14th European Conf on Computer Vision, p.540-555.
https://doi.org/10.1007/978-3-319-10599-4_35

Rochan M, Wang Y, 2019. Video summarization by learning from unpaired data. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.7894-7903.
https://doi.org/10.1109/CVPR.2019.00809

Rochan M, Ye LW, Wang Y, 2018. Video summarization using fully convolutional sequence networks. Proc 15th European Conf on Computer Vision, p.358-374.
https://doi.org/10.1007/978-3-030-01258-8_22

Shen T, Zhou TY, Long GD, et al., 2018. Bi-directional block self-attention for fast and memory-efficient sequence modeling. Proc 6th Int Conf on Learning Representations, p.1-18.

Song YL, Vallmitjana J, Stent A, et al., 2015. TVSum: summarizing web videos using titles. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.5179-5187.
https://doi.org/10.1109/CVPR.2015.7299154

Szegedy C, Liu W, Jia YQ, et al., 2015. Going deeper with convolutions. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1-9.
https://doi.org/10.1109/CVPR.2015.7298594

Wei HW, Ni BB, Yan YC, et al., 2018. Video summarization via semantic attended networks. Proc AAAI Conf on Artificial Intelligence, p.216-223.

Yu F, Koltun V, 2016. Multi-scale context aggregation by dilated convolutions. http://arxiv.org/abs/1511.07122

Yuan L, Tay FE, Li P, et al., 2019. Cycle-SUM: cycle-consistent adversarial LSTM networks for unsupervised video summarization. Proc AAAI Conf on Artificial Intelligence, p.9143-9150.
https://doi.org/10.1609/aaai.v33i01.33019143

Yuan YT, Mei T, Cui P, et al., 2019. Video summarization by learning deep side semantic embedding. *IEEE Trans Circ Syst Video Technol*, 29(1):226-237.
https://doi.org/10.1109/tcsvt.2017.2771247

Zhang K, Chao WL, Sha F, et al., 2016. Video summarization with long short-term memory. Proc 14th European Conf on Computer Vision, p.766-782.
https://doi.org/10.1007/978-3-319-46478-7_47

Zhao B, Xing EP, 2014. Quasi real-time summarization for consumer videos. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.2513-2520.
https://doi.org/10.1109/CVPR.2014.322

Zhao B, Li XL, Lu XQ, 2018. HSA-RNN: hierarchical structure-adaptive RNN for video summarization. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.7405-7414.
https://doi.org/10.1109/CVPR.2018.00773

Zhao B, Li XL, Lu XQ, 2020. Property-constrained dual learning for video summarization. *IEEE Trans Neur Netw Learn Syst*, 31(10):3989-4000.
https://doi.org/10.1109/TNNLS.2019.2951680

Zhou KY, Qiao Y, Xiang T, 2018. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. Proc AAAI Conf on Artificial Intelligence, p.7582-7589.

Zhuang YT, Rui Y, Huang TS, et al., 1998. Adaptive key frame extraction using unsupervised clustering. Proc Int Conf on Image Processing, p.866-870.
https://doi.org/10.1109/ICIP.1998.723655