GIF Thumbnails: Attract More *Clicks* to Your Videos

Yi Xu[1*], **Fan Bai**[1], **Yingxuan Shi**[2], **Qiuyu Chen**[3], **Longwen Gao**[2], **Kai Tian**[1]
Shuigeng Zhou[1†], **Huyang Sun**[2]
[1]Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, China
[2]Bilibili, China
[3]Department of Computer Science, UNC Charlotte
{yxu17,fbai19,ktian14,sgzhou}@fudan.edu.cn; {shiyingxuan,gaolongwen,sunhuyang}@bilibili.com; qchen12@uncc.edu

Abstract

With the rapid increase of mobile devices and online media, more and more people prefer posting/viewing videos online. Generally, these videos are presented on video streaming sites with image thumbnails and text titles. While facing huge amounts of videos, a viewer clicks through a certain video with high probability because of its eye-catching thumbnail. However, current video thumbnails are created manually, which is time-consuming and quality-unguaranteed. And static image thumbnails contain very limited information of the corresponding videos, which prevents users from successfully clicking what they really want to view.

In this paper, we address a novel problem, namely *GIF thumbnail generation*, which aims to automatically generate GIF thumbnails for videos and consequently boost their *Click-Through-Rate* (CTR). Here, a GIF thumbnail is an animated GIF file consisting of multiple segments from the video, containing more information of the target video than a static image thumbnail. To support this study, we build the first GIF thumbnails benchmark dataset that consists of 1070 videos covering a total duration of 69.1 hours, and 5394 corresponding manually-annotated GIFs. To solve this problem, we propose a learning-based automatic GIF thumbnail generation model, which is called ***Ge****nerative* ***Va****riational* ***D****ual-****En****coder* (**GEVADEN**). As not relying on any user interaction information (*e.g.* time-sync comments and real-time view counts), this model is applicable to newly-uploaded/rarely-viewed videos. Experiments on our built dataset show that GEVADEN significantly outperforms several baselines, including video-summarization and highlight-detection based ones. Furthermore, we develop a pilot application of the proposed model on an online video platform with 9814 videos covering 1231 hours, which shows that our model achieves a 37.5% CTR improvement over traditional image thumbnails. This further validates the effectiveness of the proposed model and the promising application prospect of GIF thumbnails.

Introduction

The past decade has witnessed a phenomenal surge in video sharing. According to statistics, hundreds hours of videos are uploaded to Youtube per minute (Jamil et al. 2018). With such huge amounts of videos, whether a certain video will be clicked and further viewed with high probability depends on its thumbnail and title. Videos with eye-catching thumbnails will definitely attract more viewers. Therefore, how to create attractive image thumbnails to boost Click-Through-Rate (*CTR*) becomes a major concern of video publishers and streaming sites (Song et al. 2016; Yuan, Ma, and Zhu 2019; Zhao et al. 2017; Kim et al. 2019).

Fig. 1 shows two different kinds of thumbnails of a video: 1) a static image (*left*) and 2) an animated GIF consisting of three shots (*right*). Obviously, the right one is more attractive because the GIF provides more vivid snapshots of the video and consequently will stimulate users' stronger interest in clicking and viewing. However, manually creating thumbnails (especially GIF thumbnails) is time-consuming and quality-unguaranteed. Thus, it is desirable to automatically generate thumbnails in the form of animated GIFs.

Nowadays, some popular video streaming sites, such as YouTube, begin to trim a short segment from a video as the thumbnail to make it more attractive. However, only one trimmed segment may lead to misunderstanding or even click-baits for limited video gist information, while too many shots in a GIF thumbnail will dampen viewers' interest and patience. For example, with only the 2nd shot in Fig. 1 as the thumbnail may make viewers mistake bullying as having fun, but more than 3 shots will cost viewers too much patience to browse the thumbnail. On the other hand, user interaction information (*e.g.* time-sync comments and real-time view counts) is helpful for locating interesting shots and generating GIF thumbnails. Nevertheless, for newly-uploaded or rarely-viewed videos, there is little user interaction information available.

With the observation above in mind, we propose a novel problem: *automatically generating GIF thumbnails for videos* from computer vision perspective. Considering that most stories can be summarized in a three-part structure ("start-develop-end"), we define a GIF thumbnail as three shots from the target video in this paper. Therefore, our task is to select three shots from the target video and rearrange them, not necessarily in the original chronological order. From visual perspective, a GIF is short, and sometimes contains unique spatio-temporal visual patterns to thread multiple shots with video gist content or of interestingness.

To support this study, we first build a benchmark dataset for GIF thumbnail generation, which consists of 1070 videos of

[*]This author did most work during his internship at Bilibili.

[†]Corresponding author.

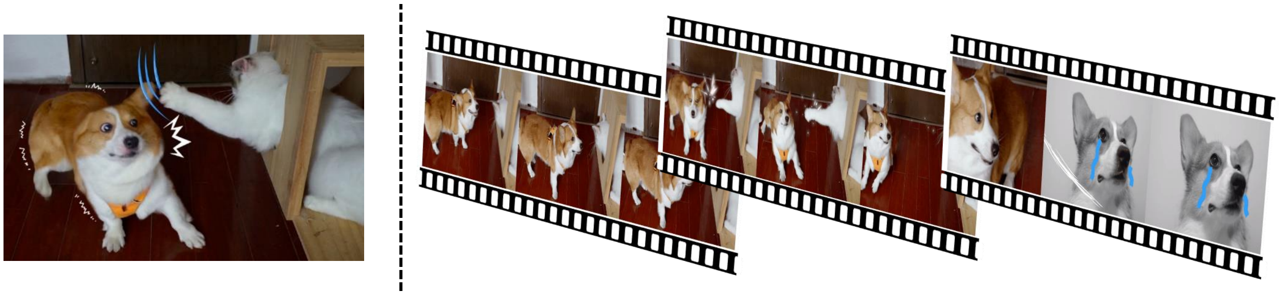


Figure 1: A video with two different kinds of thumbnails: static image (*left*) and animated GIF (*right*). The left is an image thumbnail created by the video owner, which is carefully selected to show the situation that the dog is scared to tremble and bullied by the white cat. The right includes several cut screens from three shots in an animated GIF, from which we can see the storyline of the video: the dog wants to show kindness to the cat in the 1st shot, but it is hit by the cat in the 2nd shot. Then, it cries in the 3rd shot. All these shots in the animated GIF are directly extracted from the video without modification.

various contents and a total duration of 69.1 hours. Considering the difficulty of searching the optimal GIF thumbnail for a video, we take several GIFs as the ground truth of each video. To generate GIF thumbnails, we introduce a model based on Variational Auto-Encoder (VAE) (Kingma and Welling 2014) to learn the video-dependent latent patterns of GIFs, which is called ***Ge***nerative ***Va***riational ***D***ual-***En***coder (**GEVADEN** in short). Here, GEVADEN is a combination of sequence-to-sequence model and conditional VAE. Experimental results on the benchmark verify the effectiveness of GEVADEN and its advantage over the competing methods. Furthermore, we deploy a pilot application of our model on a real video streaming platform with 9814 non-annotated videos. Running results of the application show that using GIF thumbnails generated by GEVADEN achieves 37.5% *CTR* improvement over using static image thumbnails, which demonstrates the promising prospect of GIF thumbnails in real applications.

In summary, contributions of this paper are as follows:

- We introduce a novel problem of automatically generating GIFs as thumbnails from vision perspective. To the best of our knowledge, this is the first work to address such a novel and challenging problem in the literature.
- To support the study of automatic GIF thumbnail generation, we build the first benchmark dataset with 1070 videos and 5394 carefully annotated GIF thumbnails.
- We develop an effective model called Generative Variational Dual-Encoder (GEVADEN) to generate GIF thumbnails by learning the latent patterns of annotated GIFs.
- We conduct extensive experiments on the benchmark and deploy a pilot application on a real video streaming platform. Results of experiments and application running validate the proposed model, and a 37.5% CTR improvement demonstrates its potential in real applications.

Note that some existing vision tasks such as image thumbnail selection (Song et al. 2016), video highlight detection (Xiong et al. 2019), and video summarization (Zhang, Grauman, and Sha 2018) are related to our problem, but they are not suitable for or effective in our scenario. On the one hand, for video summarization, though we can reduce the length of the summary to 3 shots, most of the existing methods (Mahasseni, Lam, and Todorovic 2017; Rochan, Ye, and Wang 2018; Wei et al. 2018a; Jung et al. 2019) with binary or importance score output and an additional selector (Gong et al. 2014) are not suitable for GIF thumbnail generation, because they do not consider the chronological orders of the selected shots. On the other hand, highlight detection and image thumbnail selection consider only one segment/frame in a video, while GIF thumbnail needs to select multiple shots jointly. Although highlight detection can be extended to select multiple shots via top-k ranking scores, it cannot be guaranteed that the selected shots contain video gist information and inter-shot semantic connection, which are indispensable to GIFs. Most importantly, we adapt four existing methods, including one image thumbnail selection method, one summarization method, and two highlight detection methods to generating GIF thumbnails. Experimental results show that our method outperforms these methods.

Related Work

Here we survey the latest advances in video summarization and video highlight detection areas. We also review the techniques closely related to the proposed model.

Video ummarization. It is to produce a set of shots or keyframes that cover the gist of the original video. Summaries are output in various structured forms, including storyline graphs (Kim and Xing 2014; Xiong, Kim, and Sigal 2015), montage representations (Kang et al. 2006), a sequence of key frames (Zhang et al. 2016; Mahasseni, Lam, and Todorovic 2017) or shots (Zhang, Grauman, and Sha 2018), according to different understandings and requirements of the task. In what follows, we focus on the methods that output a sequence of keyframes or clips.

As a long-standing computer vision task, video summarization considers not only the importance of frames but also the criteria of relevance, representativeness, interestingness, and diversity (Hong et al. 2011; Mei et al. 2015; Gong et al. 2014). Some (Mahasseni, Lam, and Todorovic 2017; Rochan, Ye, and Wang 2018; Wei et al. 2018a; Jung et al. 2019) output binary value (importance score) for each frame and select keyframes through DPP (Gong et al. 2014).

For methods with regression strategy, (Zhang, Grauman, and Sha 2018) generates the summary with the nearest neighbor algorithm based on the output sequence, while (Fu, Tai, and Chen 2019) predicts time stamps of selected keyframes with Ptr-Net (Vinyals, Fortunato, and Jaitly 2015). Recently, Many methods (Cai et al. 2018; Fu, Tai, and Chen 2019; Jung et al. 2019; Mahasseni, Lam, and Todorovic 2017; Yuan et al. 2019) exploit unannotated data or Web priors via weakly-supervised/unsupervised strategies.

Video Highlight Detection. It retrieves a moment that may catch viewers' attention. Early works mainly investigate broadcast sports videos (Rui, Gupta, and Acero 2000; Wang et al. 2004; Xiong et al. 2005; Tang et al. 2011). Recent studies focus on specific-domain videos (Sun, Farhadi, and Seitz 2014; Merler et al. 2017; Sun et al. 2017; Kim et al. 2018; Xiong et al. 2019), and the detected highlights come with a strong prior (category labels or keywords), like surfing and dog etc. So given a video, different highlights may be detected for different scenarios. For instance, (Kim et al. 2018) exploits domain-specific top-ranked web images as weak supervision and proposes a novel triplet deep ranking strategy. (Wei et al. 2018b) proposes a Sequence-to-Segments Network and uses Earth Mover's distance to predict the localities and scores of segments. (Xiong et al. 2019) assumes that shots in shorter videos have higher highlight score than those in longer videos, and proposes a novel video clip ranking framework for noise data. More similar to our task, Video2Gif (Gygli, Song, and Cao 2016) was proposed based on RankNet and learned from GIF-video pairs created by rules, and each GIF consists of only one segment, different from ours.

Variational Auto-Encoder (VAE). VAE (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014) has been explored in various tasks (Mahasseni, Lam, and Todorovic 2017; Cai et al. 2018; Denton and Fergus 2018). Concretely, (Denton and Fergus 2018) develops a recurrent inference network to estimate latent distribution for each time step in the frame prediction task. (Mahasseni, Lam, and Todorovic 2017) integrates VAE and generative-adversarial training into a deep architecture for unsupervised video summarization. (Cai et al. 2018) utilizes videos from Web to estimate common latent distribution via VAE.

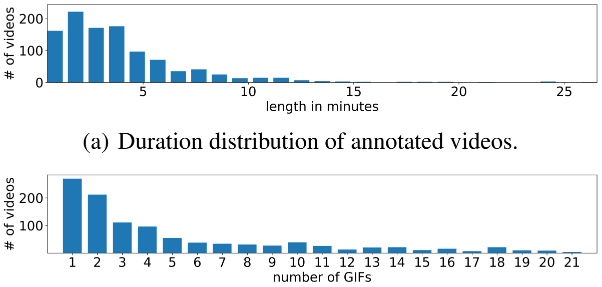
In our work, *we use VAE to learn GIF patterns as the posterior distribution from the ground-truth representations*. We merge the decoder of VAE and a video encoder into a sequence-to-sequence model for GIF thumbnail generation. As illustrated in Fig. 3, the generator of our model takes video embedding and a random vector from latent distribution to predict GIF thumbnails. The random vector for the generator is drawn from the encoder of VAE in the training phase, but from the encoder of our model in the test phase.

Problem Statement

Given a video $X = [x_1, x_2, ..., x_n]$, and a set of GIF thumbnails $T = \{T_1, T_2, ..., T_m\}$, where n is the length of video, m is the number of ground-truth thumbnail for video X, x_i indicates the feature vector of the i-th shot (in this paper, **each shot is 1-second length**), and T_j is the j-th GIF thumbnails of video X. For simplicity of notation, in the following we represent an arbitrary ground-truth GIF thumbnail T_j ($j \in \{1, 2, ..., m\}$) as $Y = [y_1, y_2, y_3]$ in the form of a sequence of 3 time stamps, each of which indicates the start time of a shot in the video. Time stamps in Y are not necessarily arranged in chronological order as aforementioned.

Dataset	Task or problem	# of videos	duration
OVP	S	50	each from 1 to 4m
Youtube	S	50	each from 1 to 10m
SumMe	S	25	1.2h
CoSUM	S	51	2.5h
TVSum	H & S	50	3.5h
YouTube Highlights	H	712	23.8h
Yahoo Screen	H	1,118	52.5h
VTW	H & S	2,000	50h
SVCD	H	11,000	each less than 10s
our dataset	GIF Thumbnails	1,070	69.1h

Table 1: Statistics comparison of our dataset with some existing commonly-used vision task datasets. "S" means video summarization, "H" indicates highlight detection. Duration of OVP (De Avila et al. 2011), Youtube (De Avila et al. 2011) and SVCD (Ren et al. 2020) means single video duration.



(a) Duration distribution of annotated videos.

(b) Histogram of the number of annotated GIF thumbnails for videos.

Figure 2: Statistics of time duration and the number of annotated GIFs for videos in our dataset.

The goal of our problem is to build a model for 1) generating a GIF $\hat{Y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3]$ as the thumbnail for each training video such that the aggregated difference between the generated GIFs and the ground-truth GIFs is minimized (*training phase*), and 2) generating proper GIF thumbnails for testing videos (*test phase*) or new-coming videos such that their CTR can be boosted (*application phase*).

GIF Thumbnails Benchmark

A major obstacle for this novel task is the lack of public datasets that consist of video and GIF thumbnail pairs, as well as posterior information (*e.g.* time-sync comments) for evaluation. To tackle this problem, we collect and annotate a dataset from scratch for this new task. To the best of our knowledge, our *GIF Thumbnail dataset* is the first one for generating GIF thumbnails from vision perspective.

Data Collection and Annotation

As it is difficult to find the optimal GIF thumbnail for each video manually, we consider any acceptable annotated GIF

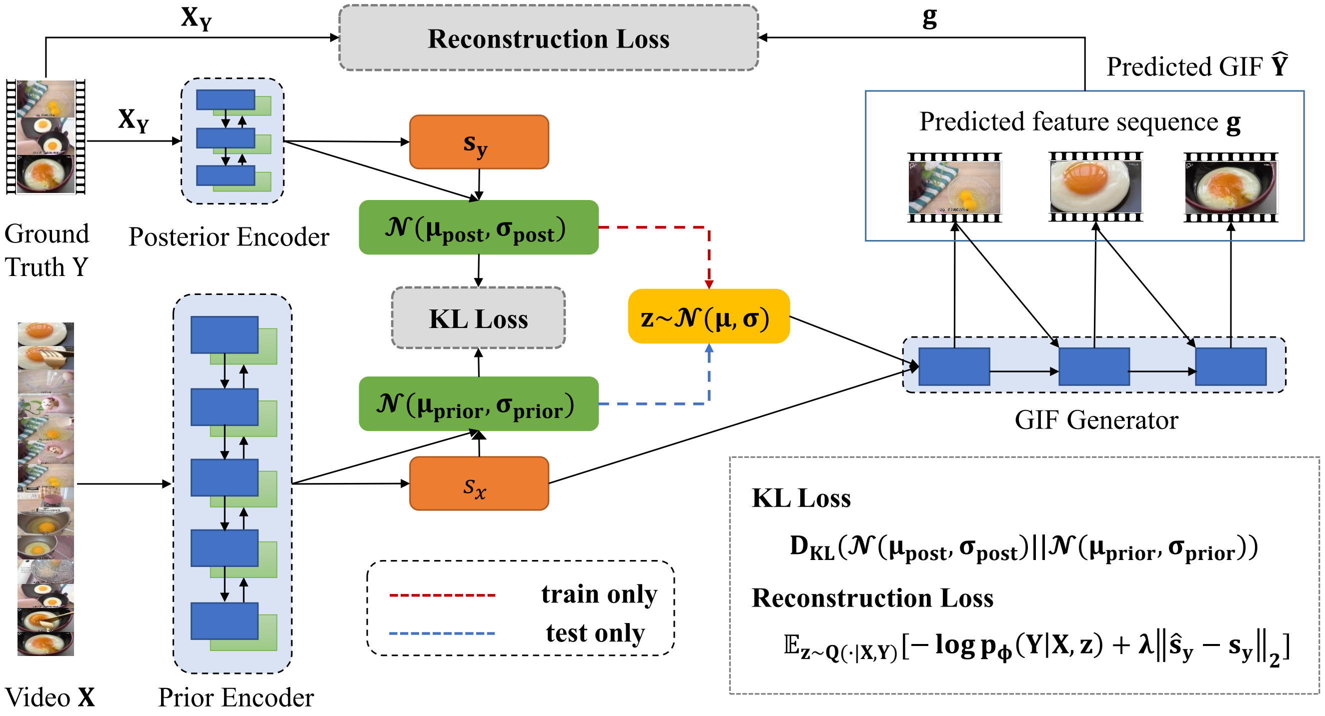

X_Y
Reconstruction Loss
g
Predicted GIF $\hat{Y}$
X_Y
s_y
Predicted feature sequence g
$\mathcal{N}(\mu_{post}, \sigma_{post})$
Ground Truth Y
Posterior Encoder
KL Loss
$z \sim \mathcal{N}(\mu, \sigma)$
$\mathcal{N}(\mu_{prior}, \sigma_{prior})$
GIF Generator
s_x
KL Loss
$D_{KL}(\mathcal{N}(\mu_{post}, \sigma_{post}) || \mathcal{N}(\mu_{prior}, \sigma_{prior}))$
Reconstruction Loss
$\mathbb{E}_{z \sim Q(\cdot|X,Y)}[-\log p_\phi(Y|X, z) + \lambda \|\hat{s}_y - s_y\|_2]$
train only
test only
Video X
Prior Encoder

Figure 3: The framework of our model. The red dashed line denotes data flow existing only in the training phase, and the blue dashed line means data flow valid only in the test phase.

as a ground-truth. Thus, each video may correspond to a certain number of GIF thumbnails, each of which consists of three disjoint shots extracted from the video. These shots are rearranged, not necessarily in their original orders.

Data Collection In the *GIF Thumbnail dataset*, videos were collected from a popular Chinese video streaming site called Bilibili. To select representative videos with non-biased user interactions, we first crawl videos uploaded three months ago to guarantee sufficient views and time-sync comments. Then, we remove low-quality or meaningless time-sync comments in videos and filter these videos via a threshold of the number of sanitized time-sync comments. The reason is that we try to keep only videos with sufficient high-quality time-sync comments in the dataset.

Annotation. Due to the large search space and its subjective nature, manually annotating GIF thumbnails is a non-trivial task, especially for long videos. Thus, we generate GIF proposals based on time-sync comments. Concretely, we first detect peaks of time-sync comments per second over the timeline. Here, a peak means the number of comments at this timestamp is larger than those of the preceding and the following timestamps. Intuitively, peak timestamps have a higher probability of being included in GIF thumbnails than non-peak timestamps, because more time-sync comments imply that more viewers are interested in the shots starting from these timestamps. We then remove the peaks located in the starting and ending segments of the videos, as shots in the starting and ending segments are possibly meaningless and unrepresentative. In such a way, we can obtain a series of representative shots for each video.

To compose a GIF, we sample three disjoint shots from the video with the probability in proportion to the number of time-sync comments contained in each shot. To create more proper thumbnails, we also apply Monte Carlo strategy and several manually-designed rules to rearrange the sampled shots, and then assemble them into a GIF proposal.

Finally, more than 30 GIF proposals are generated for each video, and each GIF is labeled by at least two annotators to indicate whether or not it is attractive to be clicked. Only these GIF Thumbnails positively labeled by all annotators are included in the dataset. Eventually, this dataset consists of 1070 videos with a time duration of 69.1 hours and 5401 corresponding GIF thumbnails.

Data Statistics

Major statistic information of our dataset is presented in Table 1. For comparison, we also give the statistic data of some commonly-used datasets for video summarization and highlight detection, including Open Video Project (OVP) (De Avila et al. 2011), Youtube (De Avila et al. 2011), SumMe (Gygli et al. 2014), CoSUM (Chu, Song, and Jaimes 2015) and TVSum (Song et al. 2015) for video summarization, YouTube Highlights (Sun, Farhadi, and Seitz 2014), Yahoo Screen (Song et al. 2016), VTW (Zeng et al. 2016) and SVCD (Ren et al. 2020) for video highlight detection. Compared to the existing datasets, our *GIF Thumbnail* *dataset* covers longer video time duration, and it also contains 9814 additional videos used in our pilot application.

Fig. 2(a) shows the distribution of video duration, ranging from 20s to 25m. Fig. 2(b) illustrates the distribution of the number of GIF thumbnails over videos, which reveals one unique feature of our problem: each video corresponds to a certain (but not fixed) number of GIF thumbnails.

Model

The major challenge of GIF thumbnail generation is how to model the relationship between a video and multiple acceptable GIF thumbnails. To this end, we propose a sequence-to-sequence model based on a Variational Auto-Encoder, which is called ***Ge***nerative ***Va***riational ***D***ual-***En***coder (***GEVADEN*** in short). In what follows, we present the framework, major modules, and the loss function of our model in detail.

Generative Variational Dual-Encoder

Framework Our model GEVADEN has an end-to-end trainable framework, as shown in Fig. 3, which consists of three major modules: *Prior Encoder*, *Posterior Encoder* and *GIF Generator*. The *Prior Encoder* is responsible for embedding videos into semantic space and learning the video-dependent prior distribution used for GIF generation in the *GIF Generator* module. The *Posterior Encoder* is used to learn unique patterns of GIF thumbnails in the latent space. These patterns are taken as the posterior distribution for model training. The *GIF Generator* generates a GIF thumbnail by combining video embedding (s_x) and a random vector z drawn from the latent space. During training, z is sampled from the posterior distribution output by *Posterior Encoder* (red dashed line in Fig. 3). While in the inference phase, z is sampled from the video-dependent prior output by *Prior Encoder* (blue dashed line in Fig. 3).

Prior Encoder We first apply a pretrained 3D-ResNet (Hara, Kataoka, and Satoh 2018) to encode each video into a feature vector sequence X. We then use a two-layer LSTM to embed the video into a semantic vector s_x, with which the prior distribution $\mathcal{N}(\mu_{prior}, \sigma_{prior})$ of latent space is characterized by a two-layer MLP with ReLU activation. Then, we draw a random sample z from the prior distribution to provide video-dependent patterns to the GIF generator for GIF thumbnail generation. Formally,

$$z \sim P(\cdot|X;\theta) = \mathcal{N}\Big(\mu_{prior}(s_x), \sigma_{prior}(s_x)\Big), \quad (1)$$

where θ indicates parameters of the two-layer LSTM and MLP, and $P(\cdot|X;\theta)$ is a Gaussian distribution with mean μ_{prior} and variance σ_{prior}.

Posterior Encoder This module is introduced to recognize the latent patterns from the ground-truth GIF thumbnails. It has a similar architecture to *Prior Encoder*. By taking feature $X_Y = [x_{y_1}, x_{y_2}, x_{y_3}]$ of Y (a ground truth thumbnail) as input, it outputs a semantic feature s_y of Y, and a recognized latent pattern mapped at the location $\mu_{post}(s_y)$ with uncertainty $\sigma_{post}(s_y)$ in the latent space. We denote the recognized latent pattern as posterior distribution $Q(\cdot|X,Y)$. A random sample z is drawn from the posterior distribution as follows:

$$z \sim Q(\cdot|X,Y;\omega) = \mathcal{N}(\mu_{post}(s_y), \sigma_{post}(s_y)) \quad (2)$$

where ω means the parameters of this module. z is then fed to the *GIF Generator* to predict a GIF thumbnail $\hat{Y}$, which is expected to be as identical as possible to the corresponding input ground-truth Y. This results in the regression penalty item in Eq. (5). Details of data flow and training strategy about this module are given in Fig. 3.

GIF Generator. The combination of *Prior Encoder* and *GIF Generator* is an extension to the sequence-to-sequence architecture. As a decoder, *GIF Generator* consists of a two-layer LSTM. Besides the video embedding feature s_x as initial hidden state, *GIF Generator* takes a random sample z from $Q(\cdot|X,Y)$ in Eq. (2) (for training) or $P(\cdot|X)$ in Eq. (1) (for inference) as input, and outputs shot features sequentially. That is,

$$g_i = G(\mathcal{H}_{i-1}, z, g_{i-1}; \phi), \quad (3)$$

where G denotes the generator, ϕ stands for the parameters of G, g_i is the i-th generated shot feature in the predicted sequence g, and $\mathcal{H}_i$ indicates the hidden state at the i-th time stamp. In Eq. (3), $\mathcal{H}_{i-1} = s_x$ and $g_{i-1} = \vec{0}$ when $i = 1$. Then, we solve the closest matching problem with a disjoint restriction to obtain the GIF thumbnail $\hat{Y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3]$:

$$\begin{aligned} \hat{Y} &= arg\min_{\hat{Y}^{(l)}} \sum_{i}^{|g|} \left\| g_i - x_{\hat{y}_i^{(l)}} \right\|, \\ s.t. &\quad \hat{y}_i^{(l)} \neq \hat{y}_j^{(l)}, \quad \forall i \neq j. \end{aligned} \quad (4)$$

Loss Function

In training stage, a sample (vector) z is drawn from the posterior distribution $Q(\cdot|X,Y)$ and fed into *GIF Generator* for GIF generation; while in the inference phase, z is drawn from the prior distribution $P(\cdot|X)$. Thus, Kullback-Leibler divergence is applied to penalizing the difference between the posterior distribution and the prior distribution. In addition, a regression loss is used to penalize the difference between the feature of the predicted GIF thumbnail g and that of ground-truth GIF thumbnail X_Y. Therefore, the optimization objective function is formulated as follows:

$$\begin{aligned} \mathcal{L}(\theta,\phi,\omega) = \mathbb{E}_{z\sim Q(\cdot|X,Y)}\Big[-\log p_\phi(Y|X,z)\Big] \\ + \beta D_{KL}\Big(Q(z|X,Y)||P(z|X)\Big), \end{aligned} \quad (5)$$

where p_ϕ is the likelihood term of regression penalty between g and X_Y. During training, the KL divergence D_{KL} pushes the posterior distribution and the prior distribution towards each other. We train the model using the re-parameterization trick (Kingma and Welling 2014), where we simulate the expectation over the posterior distribution by drawing 4 samples for each input video. The hyper-parameter β in Eq. (5) is the trade-off between minimizing the regression penalty and fitting the prior. A small β will make the model tend to under-utilize/ignore the sample z, and even deteriorate to a sequence-to-sequence model. If β is too small, the model will lack exploration due to little utilization of z and become hard to converge with multiple GIFs. If β is too large, the model tends to over-fit for the ground-truth GIFs and achieves high performance in training but low performance in testing. More discussions about hyper-parameter β can refer to the experimental part of the Appendix.

One key point of generating credible GIF thumbnails is to follow the gist of the ground-truth. Actually, only regression penalty for supervision is not enough. Inspired by perceptual loss (Johnson, Alahi, and Fei-Fei 2016), we feed the predicted feature sequence g into *Posterior Encoder* and re-embed it into the semantic space of ground-truth GIF thumbnails. We denote the re-embedded feature of the predicted GIF as $\hat{s}_y$. Thus, we have the following loss:

$$\mathcal{L}_{percept}(\theta, \phi, \omega) = \|\hat{s}_y - s_y\|_2 \,. \tag{6}$$

Note that in the perceptual loss of Eq. (6), no gradient is accumulated for ω in the re-embedding phase.

Accordingly, we have the following loss function for generating credible GIF thumbnails:

$$\mathcal{L}_{mixed} = \mathcal{L}(\theta, \phi, \omega) + \lambda \mathbb{E}_{z \sim Q(\cdot|X,Y)} \mathcal{L}_{percept}(\theta, \phi, \omega), \tag{7}$$

where λ is a trade-off parameter. With the mixed loss function above, our model can be trained via back-propagation efficiently.

Performance Evaluation

Setup

Implementation Details. We randomly split the annotated videos from the benchmark dataset into three parts: 857 for training, 106 for validation, and 107 for testing. For all videos, we resize them to 160p along the short side and resample them to 16 fps frame rate. Before applying the *Prior Encoder*, we utilize a 3D-ResNet pretrained on Kinetics (Kay et al. 2017) as a bottomed video feature extractor. The 3D-ResNet with ResNet-50 takes a 16-frame clip as input and outputs a feature vector with 2048 channels. We set the dimensions of video semantic space, GIF semantic space, and Gaussian distribution to 512, 128, and 10, respectively. We train networks via Aadm (Kingma and Ba 2014) optimizer with an initial learning rate of 3×10^{-4} and a mini-batch size of 16. Both the closest matching in *GIF Generator* and the regression penalty in our model use L_2 norm. Networks are implemented in Pytorch v1.3, and experiments are conducted on eight NVIDIA Tesla P40s. Codes and Appendix are available at https://github.com/xyiyy/GIF-Thumbnails

Baselines As our work is the first one to generate GIF thumbnails for videos, we use four related existing methods as baselines for performance comparison: BeautThumb (Song et al. 2016), GIF generation in Hecate (Hecate in short) (Song et al. 2016), RankNet (Gygli, Song, and Cao 2016) and re-SEQ2SEQ (Zhang, Grauman, and Sha 2018). BeautThumb was proposed for image thumbnail generation. To generate GIF thumbnails by BeautThumb, we first predict the time stamps of top-3 image thumbnails by analyzing the visual quality and aesthetic metrics, then thread the corresponding shots based on their time stamps to output a GIF. As for the two video highlight detection methods Hecate and RankNet, we construct GIF thumbnails with top-3 ranked shots. We directly run the source codes and follow their original settings of these three baselines[1]. Re-SEQ2SEQ is a regression-based method for video summarization. We re-implement and adapt it to the settings of GEVADEN.

Metrics. Considering that a GIF thumbnail consists of 3 independent shots, it is difficult to predict an exact matching GIF thumbnail from a long target video. Therefore, we assess the quality of generated GIF thumbnails by two specifically proposed metrics.

- **Perceptual Matching Error (PME).** Recall that we apply closest matching to generate GIFs, to exploit the properties of closest matching, we propose the Perceptual Matching Error (PME) as one of our evaluation metrics. PME is defined as the minimal distance to move the generated feature g_i so that it falls into the correct matching target. The normalized PME can be obtained by solving the following convex optimization problem:

$$\begin{aligned} PME(g_i, X, y_i) &= \min_{\vec{v}} \frac{\|\vec{v}\|}{\|g_i - x_{y_i}\|} \\ \text{s.t.} \quad \|g_i + \vec{v} - x_{y_i}\| &\leq \|g_i + \vec{v} - x_j\|, \quad \forall x_j \in X, \end{aligned} \tag{8}$$

  where $PME(g_i, X, y_i)$ denotes the normalized PME between the generated shot feature g_i and the y_i-th shot in video X, $\|\vec{v}\|$ is the L_2-norm of vector $\vec{v}$, and $\vec{v}$ is the moving vector to be searched. $PME(g, X, Y)$ can be calculated as $\frac{1}{|Y|}\sum_i PME(g_i, X, y_i)$. Intuitively, if the generated shot is located in the area where the nearest matched shot is ground-truth, then $\vec{v} = \vec{0}$ and PME is 0. Otherwise, if the generated shot is far from ground-truth, other shots may disturb the matching result when we move the generated shot along $\vec{v}$, then the result of $\vec{v}$ will tend to approach $g_i - x_{y_i}$ and PME will be close to 1. One advantage of PME is that it also considers the relationship with shots that are not ground truth, which means that feature difference between the predicted GIFs and the ground-truth GIFs is not the only factor that affects this metric.

- **Posterior Score (PS).** It is calculated with the posterior information of videos, such as time-sync comments in this work. We propose this metric based on the observation that shots with more time-sync comments usually have a higher probability of being selected to GIF thumbnails, because more time-sync comments indicate more user interactions, *i.e.*, more interesting or attractive to viewers. We aggregate the ratios of time-sync comments of selected shots as the posterior score, and the ratios are normalized with the maximal number of time-sync comments per-second. The PS value of $\hat{Y}$ is evaluated by

$$PS(\hat{Y}, C) = \sum_{i}^{|\hat{Y}|} C_{\hat{y}_i} / \max(C), \tag{9}$$

[1] BeautThumb and Hecate: https://github.com/yahoo/hecate; RankNet: https://github.com/gyglim/video2gif_code

Method	PME (val)	PS (val)	PME (test)	PS (test)
BeautThumb (Song et al. 2016)	0.267	0.906	0.268	0.938
Hecate (Song et al. 2016)	0.257	0.895	0.261	0.939
RankNet (Gygli, Song, and Cao 2016)	0.253	0.922	0.245	0.927
re-SEQ2SEQ (2018)	0.201	0.988	0.204	0.961
GEVADEN, fixed prior	0.232	0.920	0.229	0.945
GEVADEN, $\lambda = 0$	0.192	1.031	0.188	1.027
GEVADEN, $\lambda = \lambda^*$	**0.169**	**1.073**	**0.165**	**1.061**

Table 2: Comparison with baselines and ablation study on validation and test sets. λ is a hyperparameter in Eq. (7). λ^* means the optimal value tuned on the validation set.

Method	m*PV*	m*Click*	m*CTR*	Improvement
raw thumbnails	3547.0	190.0	5.30%	-
re-SEQ2SEQ	3603.8	221.4	6.36%	19.5%
GEVADEN, $\lambda = \lambda^*$	3601.2	**262.5**	**7.29%**	**37.5%**

Table 3: Running results of a pilot application with an extended test set. "raw thumbnails" means image thumbnails created by video publishers.

where C_i is the number of valid time-sync comments in the i-th shot.

Considering that the relationship between videos and GIF thumbnails in our dataset is "one-to-many", we utilize the best evaluation result among multiple GIF thumbnails as the final result of each video.

Performance Comparison and Ablation Study

Here, we compare our model GEVADEN with four baselines and conduct an ablation study by considering three versions of GEVADEN: GEVADEN with $\lambda = 0$, GEVADEN with optimal $\lambda = \lambda^*$, and GEVADEN with fixed prior. GEVADEN with fixed prior means that we use a fixed prior (standard Gaussian distribution) for z, instead of a learned data-dependent Gaussian. Fixing the prior implies that $P(\cdot|X)$ in Eq. (1) and Eq. (5) is replaced by $\mathcal{N}(0, 1)$. In evaluation, we draw 10,000 samples for each video to do Monte Carlo estimation (Sohn, Lee, and Yan 2015) on the latent space. Experimental results on both validation and test data are presented in Tab. 2.

From Tab. 2, we can see that GEVADEN with $\lambda^* = 0.1$ outperforms the other methods in terms of all evaluation metrics on both validation and test data, which validates the effectiveness of GEVADEN in GIF thumbnail generation. We also observe that GEVADEN without perceptual loss ($\lambda = 0$) achieves 7.8% and 6.9% improvement over re-SEQ2SEQ in terms of PME and PS, respectively, while the performance of GEVADEN with fixed prior is worse than that of re-SEQ2SEQ. These results show the advantage and potential of learned data-dependent prior in GIF generation. More results and analysis of fixed prior and learned prior are presented in the Appendix, which also includes additional results of the ablation study on the hyper-parameter β.

Pilot Application

To further show the effectiveness of our model and explore the potential of GIF thumbnail in practical scenarios, we deploy a pilot application of our model on a video platform. We intend to compare the application effects of static image (raw thumbnails) and GIF thumbnails in terms of CTR. Data used in the application were crawled from the Bilibili site. Totally, there are 9814 videos with time duration of 1231 hours. More application settings and results are in the Appendix.

For our model, to perform deterministic inference, we draw multiple latent vectors (z) from the learnt prior distribution $P(\cdot|X)$ and take the average of the likelihoods as follows:

$$\hat{Y}^* = \arg\max_{\hat{Y}} \frac{1}{L} \sum_{l=1}^{L} p_\phi(\hat{Y}|X, z^{(l)}),\ z^{(l)} \sim P(\cdot|X). \quad (10)$$

Given a certain kind of thumbnails (static image or GIF), we denote *PV* and *Click* the total number of times of a test video assigned to and clicked by users respectively, and m*PV*, m*Click* and m*CTR* are the mean values of *PV*, *Click* and *CTR* over the test videos. Tab. 3 summarizes the results of two methods after a month-long running of the application.

From Tab. 3, we can see that on average each video is assigned to more than 3,000 users, among which around 200 clicked the video. Compared with raw image thumbnails, using GIF thumbnails generated by re-SEQ2SEQ and GEVADEN gains significant CTR improvement, 19.5% and 37.5% ,respectively. This validates the advantage of GIF thumbnails over traditional static image thumbnails and shows a promising application prospect of GIF thumbnail generation. Further comparing the results of the two methods, we can see that our model GEVADEN clearly outperforms the baseline method re-SEQ2SEQ.

In summary, the running results of the pilot application on a real video platform show that GIF thumbnails do boost the *CTR* of videos, and the proposed model is effective in automatically generating credible GIF thumbnails and has the potential of being deployed in real applications.

Conclusion

In this paper, we introduce a novel and challenging problem, which aims to automatically generate GIFs as thumbnails for videos from vision perspective. To facilitate the study of this new problem, we build a benchmark dataset that consists of 1070 videos and 5394 corresponding annotated GIFs. We propose an end-to-end trainable model to generate multiple GIFs for each video. The proposed model outperforms the competing baselines in terms of two proposed metrics. The significant improvement of CTR in a pilot application further validates the effectiveness of our model and its promising prospect for wide application in reality. Note that this work is the first attempt to automatically generate GIF thumbnails, in the future we plan to jointly exploit visual context (videos) and textual context (titles and tags) to advance the proposed model because additional textual information should be beneficial for generating attractive and accurate GIFs.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China (NSFC) under grant No. U1636205, 2019 Special Fund for Artificial Intelligence Innovation & Development, Shanghai Economy and Information Technology Commission (SHEITC), and the Science and Technology Commission of Shanghai Municipality Project under Grant No. 19511120700.

References

Cai, S.; Zuo, W.; Davis, L. S.; and Zhang, L. 2018. Weakly-supervised video summarization using variational encoder-decoder and web prior. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 184–200.

Chu, W.-S.; Song, Y.; and Jaimes, A. 2015. Video co-summarization: Video summarization by visual co-occurrence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3584–3592.

De Avila, S. E. F.; Lopes, A. P. B.; da Luz Jr, A.; and de Albuquerque Araújo, A. 2011. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters* 32(1): 56–68.

Denton, E.; and Fergus, R. 2018. Stochastic Video Generation with a Learned Prior. In *International Conference on Machine Learning (ICML)*, 1174–1183.

Fu, T.-J.; Tai, S.-H.; and Chen, H.-T. 2019. Attentive and adversarial learning for video summarization. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1579–1587. IEEE.

Gong, B.; Chao, W.-L.; Grauman, K.; and Sha, F. 2014. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2069–2077.

Gygli, M.; Grabner, H.; Riemenschneider, H.; and Van Gool, L. 2014. Creating summaries from user videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 505–520. Springer.

Gygli, M.; Song, Y.; and Cao, L. 2016. Video2gif: Automatic generation of animated gifs from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1001–1009.

Hara, K.; Kataoka, H.; and Satoh, Y. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 6546–6555.

Hong, R.; Tang, J.; Tan, H.-K.; Ngo, C.-W.; Yan, S.; and Chua, T.-S. 2011. Beyond search: Event-driven summarization for web videos. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 7(4): 1–18.

Jamil, A.; Abdullah, M.; Javed, M. A.; and Hassan, M. S. 2018. Comprehensive Review of Challenges & Technologies for Big Data Analytics. In *2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET)*, 229–233. IEEE.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 694–711. Springer.

Jung, Y.; Cho, D.; Kim, D.; Woo, S.; and Kweon, I. S. 2019. Discriminative feature learning for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 8537–8544.

Kang, H.-W.; Matsushita, Y.; Tang, X.; and Chen, X.-Q. 2006. Space-time video montage. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 1331–1338. IEEE.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* .

Kim, G.; and Xing, E. P. 2014. Reconstructing storyline graphs for image recommendation from web community photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3882–3889.

Kim, H.; Mei, T.; Byun, H.; and Yao, T. 2018. Exploiting web images for video highlight detection with triplet deep ranking. *IEEE Transactions on Multimedia* 20(9): 2415–2426.

Kim, H.; Oh, J.; Han, Y.; Ko, S.; Brehmer, M.; and Kwon, B. C. 2019. Thumbnails for Data Stories: A Survey of Current Practices. In *2019 IEEE Visualization Conference (VIS)*, 116–120. IEEE.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations (ICLR)* .

Kingma, D. P.; and Welling, M. 2014. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Mahasseni, B.; Lam, M.; and Todorovic, S. 2017. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 202–211.

Mei, S.; Guan, G.; Wang, Z.; Wan, S.; He, M.; and Feng, D. D. 2015. Video summarization via minimum sparse reconstruction. *Pattern Recognition* 48(2): 522–533.

Merler, M.; Joshi, D.; Nguyen, Q.-B.; Hammer, S.; Kent, J.; Smith, J. R.; and Feris, R. S. 2017. Auto-Curation and Personalization of Sports Highlights Through Multimodal Excitement Measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1–9.

Ren, J.; Shen, X.; Lin, Z.; and Mech, R. 2020. Best Frame Selection in a Short Video. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning (ICML)*, 1278–1286.

Rochan, M.; Ye, L.; and Wang, Y. 2018. Video summarization using fully convolutional sequence networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 347–363.

Rui, Y.; Gupta, A.; and Acero, A. 2000. Automatically extracting highlights for TV baseball programs. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 105–115.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 3483–3491.

Song, Y.; Redi, M.; Vallmitjana, J.; and Jaimes, A. 2016. To click or not to click: Automatic selection of beautiful thumbnails from videos. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, 659–668.

Song, Y.; Vallmitjana, J.; Stent, A.; and Jaimes, A. 2015. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 5179–5187.

Sun, M.; Farhadi, A.; and Seitz, S. 2014. Ranking domain-specific highlights by analyzing edited videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 787–802. Springer.

Sun, M.; Zeng, K.-H.; Lin, Y.-C.; and Farhadi, A. 2017. Semantic highlight retrieval and term prediction. *IEEE Transactions on Image Processing (TIP)* 26(7): 3303–3316.

Tang, H.; Kwatra, V.; Sargin, M. E.; and Gargi, U. 2011. Detecting highlights in sports videos: Cricket as a test case. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. IEEE.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2692–2700.

Wang, J.; Xu, C.; Chng, E.; and Tian, Q. 2004. Sports highlight detection from keyword sequences using HMM. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, volume 1, 599–602. IEEE.

Wei, H.; Ni, B.; Yan, Y.; Yu, H.; Yang, X.; and Yao, C. 2018a. Video summarization via semantic attended networks. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*.

Wei, Z.; Wang, B.; Nguyen, M. H.; Zhang, J.; Lin, Z.; Shen, X.; Mech, R.; and Samaras, D. 2018b. Sequence-to-segment networks for segment detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 3507–3516.

Xiong, B.; Kalantidis, Y.; Ghadiyaram, D.; and Grauman, K. 2019. Less is more: Learning highlight detection from video duration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1258–1267.

Xiong, B.; Kim, G.; and Sigal, L. 2015. Storyline representation of egocentric videos with an applications to story-based search. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 4525–4533.

Xiong, Z.; Radhakrishnan, R.; Divakaran, A.; and Huang, T. S. 2005. Highlights extraction from sports video based on an audio-visual marker detection framework. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 4–pp. IEEE.

Yuan, L.; Tay, F. E.; Li, P.; Zhou, L.; and Feng, J. 2019. Cycle-SUM: cycle-consistent adversarial LSTM networks for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 9143–9150.

Yuan, Y.; Ma, L.; and Zhu, W. 2019. Sentence specified dynamic video thumbnail generation. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 2332–2340.

Zeng, K.-H.; Chen, T.-H.; Niebles, J. C.; and Sun, M. 2016. Generation for user generated videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 609–625. Springer.

Zhang, K.; Chao, W.-L.; Sha, F.; and Grauman, K. 2016. Video summarization with long short-term memory. In *European conference on computer vision*, 766–782. Springer.

Zhang, K.; Grauman, K.; and Sha, F. 2018. Retrospective encoders for video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 383–399.

Zhao, B.; Lin, S.; Qi, X.; Zhang, Z.; Luo, X.; and Wang, R. 2017. Automatic generation of visual-textual web video thumbnail. In *SIGGRAPH Asia 2017 Posters*, 1–2.