Find Objects and Focus on Highlights: Mining Object Semantics for Video Highlight Detection via Graph Neural Networks

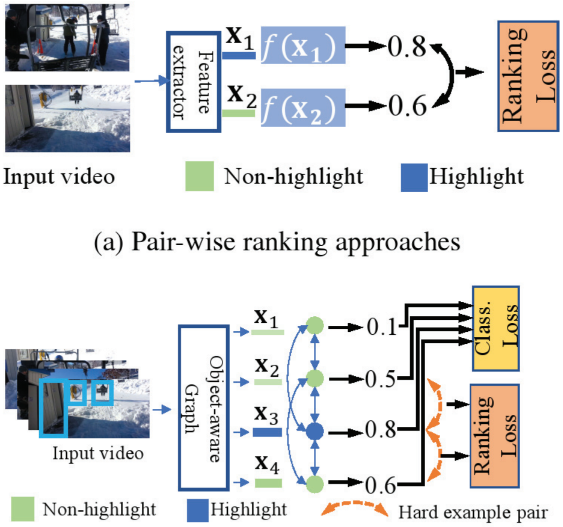
Yingying Zhang,[1,2*] **Junyu Gao,**[1,2,3*] **Xiaoshan Yang,**[1,3] **Chang Liu,**[4]
Yan Li,[4] **Changsheng Xu**[1,2,3]
[1]National Lab of Pattern Recognition,Institution of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Peng Cheng Laboratory, [4]Kuaishou Technology
{zhangyingying2017, gaojunyu2015}@ia.ac.cn, {xiaoshan.yang, csxu}@nlpr.ia.ac.cn
{liuchang03, liyan}@kuaishou.com

Abstract

With the increasing prevalence of portable computing devices, browsing unedited videos is time-consuming and tedious. Video highlight detection has the potential to significantly ease this situation, which discoveries moments of user's major or special interest in a video. Existing methods suffer from two problems. Firstly, most existing approaches only focus on learning holistic visual representations of videos but ignore object semantics for inferring video highlights. Secondly, current state-of-the-art approaches often adopt the pairwise ranking-based strategy, which cannot enjoy the global information to infer highlights. Therefore, we propose a novel video highlight framework, named **VH-GNN**, to construct an object-aware graph and model the relationships between objects from a global view. To reduce computational cost, we decompose the whole graph into two types of graphs: a spatial graph to capture the complex interactions of object within each frame, and a temporal graph to obtain object-aware representation of each frame and capture the global information. In addition, we optimize the framework via a proposed multi-stage loss, where the first stage aims to determine the highlight-probability and the second stage leverage the relationships between frames and focus on hard examples from the former stage. Extensive experiments on two standard datasets strongly evidence that VH-GNN obtains significant performance compared with state-of-the-arts.

Introduction

The video overload problem is intensifying. With the increasing prevalence of digital devices and the rapid development of social media platforms (like Taptap and Kwai), it is seamless for users to record massive amounts of videos. To mitigate the overload, *video highlight detection* has attracted increasing attention in research and industrial communities. The goal of video highlight detection is to retrieve a short video clip that captures a person's primary attention or interest within an unedited video. It also has a wide range of applications such as video retrieval, event recognition, and video recommendation. Despite much progress has been achieved in recent years (Yao, Mei, and Rui 2016; Jiao et al. 2018; Zhao, Li, and Lu 2018), this topic remains difficult due to the challenging nature of videos such as a huge semantic gap between visual features and high-level semantics, and complex temporal structures.



(a) Pair-wise ranking approaches

(b) Proposed method

Figure 1: *Comparison between our method and conventional pair-wise highlight prediction method.* (a) Pervious methods with pair-wise ranking. (b) Our VH-GNN builds an object-aware graph to capture the object semantics and the global information with a proposed multi-stage loss.

Video highlight detection algorithms are generally categorized as either unsupervised or supervised methods. Unsupervised techniques create video highlights by employing heuristics, such as video duration (Xiong et al. 2019) and visual co-occurrence (Chu, Song, and Jaimes 2015), to achieve desired characteristics. Without human-guided signals, however, the results are not satisfying enough. As opposed to unsupervised ones, supervised approaches explicitly make full use of the correspondence between the predicted video highlight and the human-annotated one, which achieve promis-

[*]Equal Contributions.

ing performance. Therefore, supervised methods have recently obtained significant attention.

To learn video highlight in a supervised fashion, as shown in Figure 1(a), current state-of-the-art methods (Yao, Mei, and Rui 2016; Jiao et al. 2018; Xiong et al. 2019) mainly utilize a pair-wise ranking constraint for two video segments with a contrastive relationship. Although these methods achieve promising results, they suffer from two problems: **(1)** Most existing approaches only focus on learning holistic visual representations of video segments but ignore object semantics for inferring video highlights. In fact, recent studies have shown that object semantics possess the remarkable ability for various video understanding tasks such as video classification (Jain, Van Gemert, and Snoek 2015; Wang and Gupta 2018) and video reasoning (Baradel et al. 2018). Note that we observe that video segments with rich objects and object-object interactions are more likely to be highlight than B-roll footage. For example, in *surfing* videos, frames with people on the surfboard are more likely to be highlight than the dull quiet ocean. As a result, video highlight detection could get benefit from object semantics. **(2)** The pairwise ranking-based methods cannot enjoy the global information of an unedited video. Intuitively, humans often take the long-range context of a video segment into consideration and select highlights from a global perspective. Note that, different frames with the same semantic content may have varying highlight strengths in different videos. For instance, frames of drinking water should be determined as a highlight in a water advertisement, but it is non-highlight in an academic lecture video. Although RNN-based approaches can be applied for video highlight detection with temporal modeling (Zhao, Li, and Lu 2018), a limitation is that they can hardly capture the temporally non-consecutive and long-distance relationships among semantics. Obviously, it is difficult for RNNs to directly model relationships between any two video frames. Besides, training RNNs may suffer from various problems and the training speed is not promising (Vaswani et al. 2017).

Recently, *Graph Neural Networks (GNNs)*, which can model the dependencies and propagate messages between any two nodes in an arbitrary graph, have received increasing attention in video understanding (Wang and Gupta 2018). GNNs possess great potential in modeling object-level semantic interactions and global associations. By explicitly constructing edges between objects, graph-based approaches can capture such information for video highlight detection. Besides, the training speed of graph-based approaches is faster than RNN-based models since GNNs do not rely on sequential operations. Until now, the application of GNNs to video highlight detection is yet to be explored.

Motivated by the above observations, as shown in Figure 1(b), we propose a novel video highlight framework, named **VH-GNN**, to directly model the object-level interactions among video frames via graph neural networks. Here, to model the spatial and temporal relationships between objects, we can build a global graph that connects all the objects (graph nodes) within the whole video. However, learning relationships across all the objects suffers from the high computational burden. To reduce the computational cost, as shown in Figure 2, we further decompose the whole graph into two types of graphs: **(1)** We design a series of spatial graphs to model the complex relations of objects within each frame independently. Specifically, for each frame in the video, we use a pre-trained object detector to extract features of each object. Since all the objects may have interactions under various scenes, we adopt a fully-connected graph to depict the spatial relationships. Because the relationships between objects are asymmetric, we design the spatial graph as a directed graph. **(2)** After the graph operation on each spatial graph, we adopt a pooling strategy to obtain the object-aware representations of each frame. These representations can be organized as a temporal graph to capture the interactions between video frames from a global view. We also formulate it as a directed graph to model the asymmetry in the temporal domain. For the graph operator design, most existing GNNs suffer from the mixing problem (Li, Han, and Wu 2018), i.e., the features of vertices will converge to the similar value. To overcome this problem, we consider utilizing edge features to update the representations of graph nodes. We optimize the model via a proposed multi-stage loss, where the first stage aims to determine the highlight-probability of each frame via a classification loss and the second stage adopts a ranking loss to leverage the relationships between frames and focus on the hard examples from the former stage. Extensive experiments on two popular datasets demonstrate the favorable performance against state-of-the-art methods. We even obtain an absolute gain of 11% on the SumMe (Gygli et al. 2014) dataset.

The main contributions of this paper are as follows:

- We propose a novel GNN-based framework that can effectively model the object semantics and global information for video highlight. To the best of our knowledge, our method is among the first to advance graph neural networks and object semantics for video highlight detection.
- By carefully designing a decomposed spatial-temporal graph, the proposed method achieves favorable computational efficiency and alleviate the overfitting problem. Here, the spatial graph can model the object semantics within each frame and the temporal graph can leverage the global relationships. Besides, our graph operator is suitable for handling the mixing problem of GNNs.
- Different from current methods that use either classification loss or ranking loss for video highlight detection, we design a novel multi-stage loss to combine both loss functions to improve the discriminative ability of our model.

Related Work

Video Highlight Detection

Video highlight detection aims to score individual video segments for their worthiness as highlights. In general, there are two main research lines: unsupervised and supervised.

Unsupervised video highlight detection methods can be further divided into methods that are domain-agnostic or domain-specific. As for domain-agnostic approach, Mendi *et al*. propose motion strength (Mendi, Clemente, and Bayrak 2013) that operates uniformly on any video.

Domain-specific approaches tailor highlights to the topic domain, and leverage video duration (Xiong et al. 2019) and visual co-occurrence (Chu, Song, and Jaimes 2015) as the weak supervision signal, or leverage category-aware reconstruction loss (Yang et al. 2015a). However, without human-guided signals, the results are not satisfying enough.

On the other hand, supervised methods treat video highlight detection as the classification or ranking task. Some methods focus on selecting keyframes or segments independently (Wang et al. 2017; Zhang et al. 2018), and treat the highlight detection as a binary classification task. However, these approaches ignore the relationship between segments and lost a lot of information. Yao *et al*. propose DCNN and employ deep learning techniques to learn the relationship between highlight and non-highlight video segments with a pairwise deep ranking model (Yao, Mei, and Rui 2016).

However, the approaches mentioned above do not make full use of the object semantics and global information of the whole video. In this work, we build a object-aware graph and decompose the whole graph into two types of graphs, a spatial graph to capture the object semantics and a temporal graph to model the global information.

Graph Neural Network

In recent years, generalization of neural networks for arbitrarily structured graphs has drawn considerable attention (Wu et al. 2019b; Zhang, Cui, and Zhu 2018; Zhou et al. 2018b; Gao, Zhang, and Xu 2019a; Gao et al. 2017). The convolution operation can be applied in the spatial or spectral domain. In the spatial domain, the methods apply feed-forward networks to each node of the graph (Scarselli et al. 2009; Li et al. 2016), and iteratively propagate nodes in the graph until the nodes reach a stable fixed point. In the spectral domain, several approaches design localized operators on graphs via convolution theorem (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2016).

Until now, graph neural networks have shown great potential in many video-related tasks. For example, group activity recognition (Deng et al. 2016; Qi et al. 2018; Wu et al. 2019a), relation modeling (Sun et al. 2019) and relation reasoning (Zhou et al. 2018a). Wang *et al*. (Wang and Gupta 2018) propose to interpret videos as space-time region graphs which consider similarity relationships and spatial-temporal relationships. Gao *et al*. (Gao, Zhang, and Xu 2019b) design a two-stream GCN model for zero-shot action recognition with relationship modeling.

Object Semantics for Video Understanding

Complex videos like human actions and events have been shown to strongly relate to their involved objects, which provide rich context information for video understanding (Marszałek, Laptev, and Schmid 2009; Wu et al. 2016; Yang et al. 2015b; Yang, Zhang, and Xu 2016; Gao, Zhang, and Xu 2017; Yang, Zhang, and Xu 2014). Sun *et al*. utilize an LSTM network as a temporal model with considering high-level object semantic features (Sun et al. 2016). Ma *et al*. learn to model higher-order object interactions between arbitrary subgroups of objects for various video understanding tasks, such as video captioning (Ma et al. 2018). Until now, the application of object semantics for video highlight detection is rarely explored.

Methodology

In this section, we elaborate on our model for video highlight detection. We first give the problem definition. Then we introduce the video graph representation and present how to operate on video graphs. Finally, we introduce our multi-stage loss function for model training.

Problem Definition

For each video, the provided annotations are a set of sampled frames and their labels $Y = \{y_1, ..., y_M\}$, each of which describes the frame is highlight(1) or not(-1). We denote the total number of sampled frames as M. What we want to do is to predict the highlight score of the sampled frames.

Graph Representation for Videos

Object Feature Extraction To capture the object semantics within a frame, we extract the object features by a pre-trained object detector. Given a video, we apply the Region Proposal Network (RPN) (Ren et al. 2015) to generate object bounding boxes on each frame. Taking the video features and projected bounding boxes, we apply RoIAlign (He et al. 2018) to extract the feature of each object region.

Spatial-Temporal Graph To capture the object-level interaction, we can build a global graph that connects all the objects (graph nodes) within the whole video. However, performance may suffer due to the fact that a finite-capacity neural network is used to model a large combinatorial space. To reduce the high computational burden, we decompose the whole graph into two types of graphs, a series of spatial graphs within the frames and a temporal graph among the frames. Here, the spatial graphs can be regarded as graph signals inputted to the temporal graph, as shown in Figure 2.

The spatial graph models the complex relations of objects and the whole image frame. Here, we use the feature of the whole image to capture other useful information (e.g., scene, background) except for objects. For the t-th sampled frame, the spatial graph can be defined as $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, W^s)$, where $\mathcal{V}_t$ is a finite set of vertices, corresponding to the $N-1$ object region proposals and one node of the whole image, as shown in Figure 2. $\mathcal{E}_t$ is a set of edges, because the relationships between objects are asymmetric, we design the spatial graph as a directed graph, and all vertices are connected to each other; W^s denotes the input-to-hidden weight matrix, which is shared among the frames.

The temporal graph models the global information of the video and the interaction between video frames. We utilize the global information because humans take the long-range context of a video segment into consideration and select highlights from a global perspective. The temporal graph can be denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W^t)$, where elements in $\mathcal{V}$ corresponding to the sampled frames; The edges in $\mathcal{E}$ exist in every two frames; W^t denotes the weight matrix of $\mathcal{G}$.

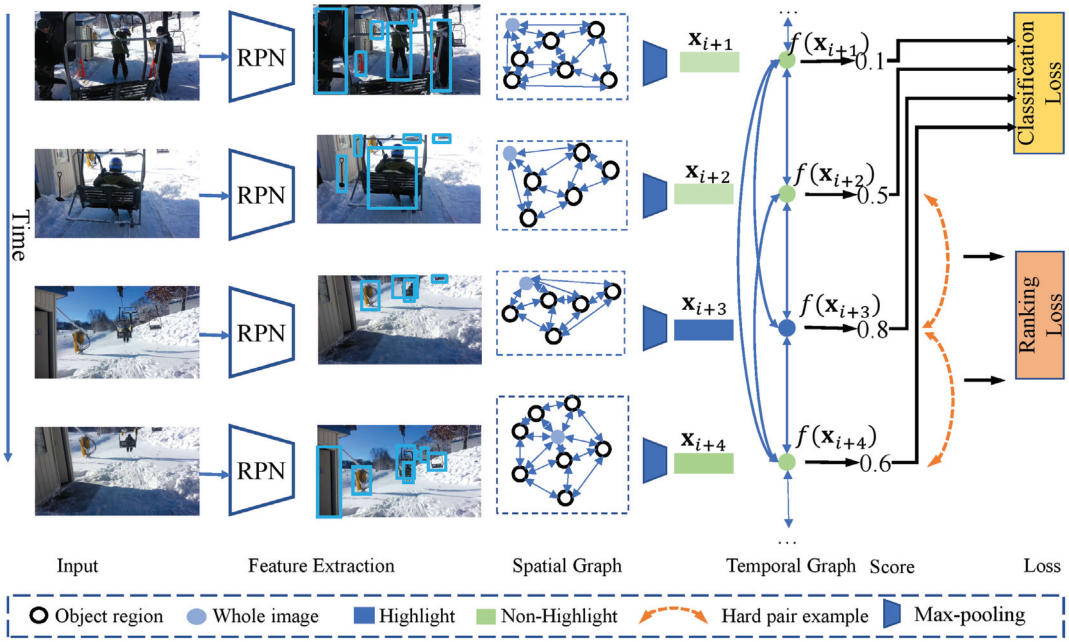


Figure 2: Overview of the proposed VH-GNN. The input is the sampled frames. We first use a RPN to extract the object region proposals and features. Then we construct a spatial graph for each frame and apply message passing over the graph, where each nodes corresponding to one object region or the whole image. Note that all the nodes are fully-connected, however, to avoid line crossing, we omit some edges. After the graph operations on spatial graphs, we use max-pooling to generate the object feature in each frame, and construct the temporal. Finally, we optimize the model via a proposed multi-stage loss.

Graph Neural Network on Video Graphs

In this section, we present the graph operator that is applied to the spatial and temporal graphs. We use the same graph operator on both types of video graphs.

Edge Feature To capture the complex relations between graph nodes, we consider utilizing edge features to update their representations. The features are calculated from the source and target nodes of the edge. We use a two-layer fully-connected network H with the hidden layer size of d_1:

$$\mathbf{e}_{i,j} = H(\mathbf{x}_i \| \mathbf{x}_j), \tag{1}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the source and target node features of the edge $\mathbf{e}_{i,j}$, and $\mathbf{e}_{i,j}$ is a d_2-dimensional vector. $\|$ denotes the concatenating operation.

Message-Passing Graph operators allow us to compute the response of a node based on its neighbors defined by the graph relations, which is equal to performing message passing inside the graph. We use a message function $M(\cdot)$ to aggregate information from all neighbors of each node with the features of edges. In particular, for each node, the message is defined as below,

$$\mathbf{m}_i = \sum_{j, j \neq i} \alpha_{i,j} M(\mathbf{x}_j \| \mathbf{e}_{i,j}), \tag{2}$$

where $M(\cdot)$ is a two-layer fully-connected neural network with the hidden layer size of d_3, $\alpha_{i,j} = \sigma((\mathrm{ReLU}(\mathbf{a}_l \cdot \mathbf{x}_i + \mathbf{a_r} \cdot \mathbf{x}_j)))$ is the attention weight, where $\mathbf{a}_l$ and $\mathbf{a}_r$ are two learnable d_1-dimensional vectors and σ is the softmax function. We now need to define a mechanism that utilizes the message received from the node's neighbors and its previous state to update its state. Therefore, we use a residual layer to update the feature of the node in each time step.

$$\mathbf{x}_i = \hat{\mathbf{x}}_i + \mathbf{m}_i, \tag{3}$$

where $\hat{\mathbf{x}}_i$ is the original feature of node i.

Comparing with traditional graph neural networks which averages the neighbors of a node for learning node representations, our graph operator leverages edge features to update the representations of graph nodes. Since $\mathbf{e}_{i,j} \neq \mathbf{e}_{j,i}$ and the messages $\mathbf{m}_i$ are significantly different among nodes, the mixing problem is alleviated.

Highlight Prediction

As shown in Figure 2, to predict if the frame is highlight or not, the graph operations are firstly conducted on each spatial graph. Then, a max-pooling strategy is performed to obtain the object-aware representation (a feature vector) of each frame as the representation for each $\mathcal{G}_t$. These representations are fed to the temporal graph as node features. Finally, we apply graph operations on the temporal graph and calculate the highlight score for each frame(nodes in the temporal graph) with a feed-forward layer:

$$f(\mathbf{x}_i) = W\mathbf{x}_i + b, \tag{4}$$

where $\mathbf{x}_i$ is the output feature of each node in the temporal graph, and $f(\mathbf{x}_i)$ is the highlight score for the frame.

Loss Function

We optimize the whole framework via a proposed multi-stage loss. The first stage is a classification loss $\mathcal{L}_c$, which aims to determine the highlight-probability of each frame:

$$\mathcal{L}_c = CrossEntropy(f(\mathbf{x}_i), y_i). \tag{5}$$

The second stage adopts a ranking loss $\mathcal{L}_r$ to leverage the relationships between frames and focus on the hard examples from the former stage, which is defined as follows:

$$\mathcal{L}_r = \frac{1}{|\mathcal{Z}|} \sum_{(p,n)\in\mathcal{Z}} \max\left(0, 1 - f\left(\mathbf{x}_p\right) + f\left(\mathbf{x}_n\right)\right), \tag{6}$$

where $\mathcal{Z}$ is the hard sample set, in which each pair (p, n), consists of a highlight frame p and a non-highlight frame n, is constructed by the following steps:

1. Sort the frames by their scores obtained in the first stage;
2. Select $r\%$ frames with lowest scores for positive labeled samples, and denoted the set of frames as $\mathcal{P}$, select $r\%$ highest scores for negative labeled samples, and denoted the set of frames as $\mathcal{N}$, where $r\%$ is set to 80%;
3. $\mathcal{Z}$ is the set of paired frames, for the items in each pair, one is the positive example and the other is the negative example, $\mathcal{Z} = \{(p, n)|p \in \mathcal{P}, n \in \mathcal{N}\}$.

Therefore, the entire loss function is defined as follows:

$$\mathcal{L} = \lambda\mathcal{L}_c + (1-\lambda)\mathcal{L}_r + \gamma||\Theta||_F, \tag{7}$$

where λ and γ are the trade-off hyperparameters and Θ is all the learnable parameters.

Experiment

We evaluate the performance of VH-GNN against several state-of-the-art methods on two public datasets.

Dataset

Details of the two datasets are illustrated as follows.

YouTube dataset (Sun, Farhadi, and Seitz 2014): This dataset contains about 490 videos of six categories. The video is annotated as segment-level with three classes: 1-highlight; 0-normal; -1-non-highlight. Each segment includes approximately 100 frames.

SumMe dataset (Gygli et al. 2014): This dataset consists of 25 videos with different events. The video is annotated with frame-level score: 1-highlight/0-non-highlight. We follow the preprocessing procedure in (Jiao et al. 2018). We treat 50 frames as a clip, and it is highlight only if the average score is higher than 0.4. We define 25 tasks for training and test, where each task preforms highlight prediction for one video and uses the remaining 24 videos as training data.

Implementation Details

Feature Extraction These two datasets have segment-level annotations. To improve computational efficiency, our model adopts sampled frames as input. We sample intra pictures (I-frames) with FFmpeg[1]. For those segments without I-frames, we sample the center frame from the segments. We use RPN with ResNet50 backbone, which is pretrained on the MSCOCO object detection dataset (Lin et al. 2014).

Parameter Setting We implement our model with Pytorch 1.1.0. For spatial graph, the number of nodes N is set to 20. The input feature dimension for each region and the whole image is 4096. The size of hidden layers d_1, d_3 and the feature size of graph nodes and edges d_2 are set to 512. We use one-layer graph neural network for spatial and temporal graph, since we do not observe much gain by adding more layers above. Since the lengths of videos are different, in each training step, we simply consider one video. Dropout is applied after each fully-connected layer with a ratio of 0.3. The balance term λ that controls the classification loss and ranking loss is set to 0.5 and the learning rate is set to 1e-5. As for weight decay, γ is setting to 0.001. Details of our implementation can be find in the opened code.[2]

Evaluation Metrics Since the two datasets have segment-level labels, during the evaluation, if a segment has more than one sampled frames, we use the average score of the frames in this segment as its highlight score. We use the mean average precision(**mAP**) as evaluation metrics, as in the most existing methods (Jiao et al. 2018; Gygli, Song, and Cao 2016; Yao, Mei, and Rui 2016)

Comparing Methods

We compare VH-GNN with four state-of-the-art methods: (1) **GIFs** (Gygli, Song, and Cao 2016), a domain-agnostic method that is trained with human-edited video-GIF pairs. (2) **LR**(Sun, Farhadi, and Seitz 2014). It introduces latent variables to accommodate highlight detection and uses the EM-like self-paced model selection procedure to train the framework effectively. (3) **DCNN** (Yao, Mei, and Rui 2016), which uses a convolutional network to extract features and detect highlight by a ranking network. To aggregate the CNN features to obtain highlight scores, they use two strategies: average-pooling (**DCA**) and max-pooling(**DCM**). (4) **AFM** (Jiao et al. 2018), which uses a fully connected layer to learn the 3D attention weights and performs max-pooling for the features of a video segment both spatially and temporally in a proposed attention module.

Quantitative Analysis

Performance Comparison From the results in Table 1– 2, we have the following observations: *YouTube Dataset*: Table 1 summarizes the overall highlight detection results for different methods on YouTube dataset (Sun, Farhadi, and Seitz 2014) and shows that our proposed approach significantly outperforms the state-of-the-art methods in most domains. In particular, the accuracies of "parkour" and "skiing" achieve 0.83 and 0.69, which makes considerable improvements over others, as they do not utilize object semantic and relationship modeling techniques in video highlight detection. Instead, we use a spatial graph to model object semantics within the frames and a temporal graph to

[1] https://ffmpeg.org/

[2] https://github.com/GNN-VH/GNN-VH

Table 1: Results comparison on the YouTube dataset.

Class	GIFs	LR	DCA	DCM	AFM	ours
gymnastics	0.34	0.40	**0.75**	0.52	0.56	0.66
parkour	0.54	0.61	0.54	0.71	0.75	**0.83**
skating	0.55	0.62	0.66	0.64	0.68	**0.70**
skiing	0.33	0.36	0.6	0.61	0.64	**0.69**
surfing	0.54	0.61	0.65	0.73	**0.78**	0.69
dog	0.31	0.60	0.58	0.69	**0.72**	0.67
Average	0.46	0.53	0.63	0.65	0.68	**0.69**

model global information, with a multi-stage loss to optimize our model. We notice that, in the "gymnastics" category, our result does not reach our expectation, and is worse than DCA. After analyzing the videos carefully, we find that those videos are mostly recorded indoors, and the scenes are similar. It is the motion of the athletes that contributes to the highlight detection significantly. However, our object-semantic based method ignores to capture such information. In the future, we will introduce the motion/action information into our framework to further improve the performance. *SumMe Dataset*: Table 2 shows the evaluation results of all 25 videos with cross validation. It is easy to see that our proposed method outperforms the comparing methods in most videos especially in "Bearpark climbing", "Cooking", "Paluma ball". On average, the average accuracy of our proposed model is 0.73, which obtains an absolute gain of 11% over the state-of-the-art method AFM.

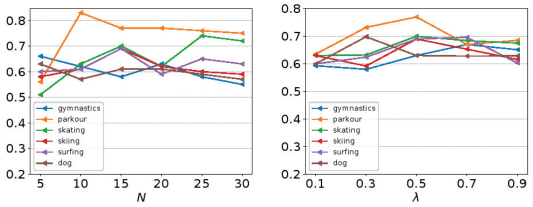
Another advantage is that we only use very few frames in one segment, but achieve comparable/superior results against baseline methods that use all of the frames. It averagely takes 1.48s for our model to predict the highlight score per video. The results demonstrate that the proposed model can achieve favorable performance with high efficiency.

Parameter Sensitivity We investigate the influence of the number of nodes in each frame N, and the trade-off term for multi-stage loss λ. We vary N from 5 to 30, while keeping other parameters fixed. The results on YouTube are presented in Figure 3(a). We can observe that, with the increase of N, the performance is boosted at first, since our model learns richer object semantics. However, the accuracy is dropped after reaching a high score due to the noises are introduced when considering more object proposals. We vary λ from 0.1 to 0.9, while keeping other parameters fixed. The results on YouTube datasets are present in Figure 3(b). We can find that for most categories, the accuracy is higher when we cooperate with the classification loss and ranking loss, which indicates our multi-stage loss strategy is useful.

In addition, to analyze the proposed model's sensitivity of the backbone network, we change the backbone from ResNet50 to AlexNet. The AlexNet is pretrained on the ImageNet database (Deng et al. 2009). For this baseline, we achieve the average mAP of 0.70 on the SumMe dataset, with 8% increment compared with AFM (Jiao et al. 2018), and only 3% decrease compared with the ResNet50. The results show that our VH-GNN is robust to various backbones.

Table 2: Results comparison on the SumMe dataset.

VideoName	DCA	DCM	AFM	ours
Air Force One	0.68	0.74	0.67	**0.80**
Base Jumping	0.64	**0.68**	**0.68**	0.60
Bearpark climbing	0.73	0.36	0.68	**0.98**
Bike Polo	0.59	0.65	**0.79**	0.68
Bus in Rock Runnel	0.57	0.47	0.65	**0.71**
car over camera	0.55	0.86	**0.95**	0.92
Car railcrossing	0.54	0.30	0.40	**0.70**
Cockpit Landing	0.51	0.60	**0.74**	0.70
Cooking	0.37	0.47	0.38	**0.78**
Eiffel Tower	0.28	0.48	0.54	**0.69**
Excavators river crossing	0.65	0.50	0.55	**0.67**
Fire Domino	0.67	0.68	0.69	**0.77**
Jumps	0.32	0.45	0.50	**0.63**
Kids playing in leaves	0.30	0.36	0.23	**0.96**
Notre Dame	0.68	0.46	0.47	**0.70**
Paintball	0.63	**0.82**	0.72	0.69
Paluma Jump	0.49	0.50	0.66	**0.86**
playing ball	0.52	0.65	0.60	**0.72**
playing on water side	0.01	0.77	0.91	**0.98**
Saving dolpines	0.46	0.12	0.12	**0.79**
Scuba	0.75	0.72	**0.95**	**0.95**
St Maarten Landing	0.78	**0.87**	0.82	0.55
Statue of Liberty	0.65	0.23	0.26	**0.93**
Uncut Evening Flight	0.49	0.68	**0.83**	0.76
Valparaiso Downhill	0.40	0.87	**0.94**	0.74
Average	0.53	0.57	0.62	**0.73**



(a) (b)

Figure 3: Parameter analysis with different N and λ.

Table 3: Accuracy of different variants on Youtube Dataset.

Class	w/o TG	w/o SG	w/o Att	FULL
gymnastics	0.54	0.63	0.63	**0.66**
parkour	0.71	0.70	0.63	**0.77**
skating	0.64	0.48	0.60	**0.83**
skiing	0.64	0.56	0.60	**0.69**
surfing	0.63	0.56	0.61	**0.69**
dog	0.63	0.63	0.57	**0.67**
Average	0.63	0.57	0.61	**0.69**

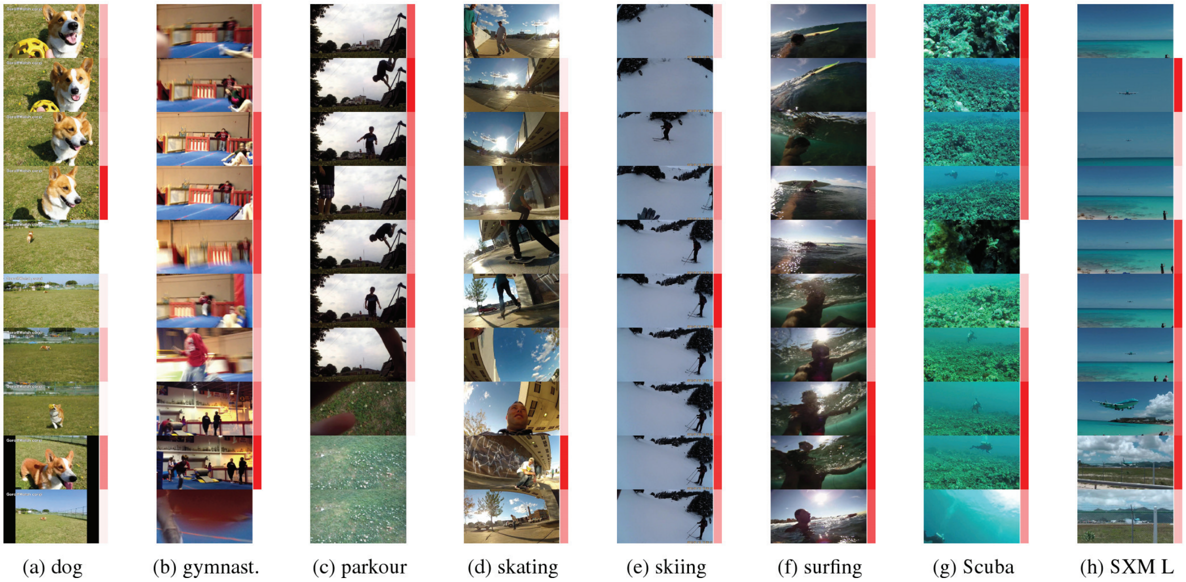


Figure 4: Examples of highlight frames and scores for videos in YouTube((a)–(f)) and SumMe dataset((g)–(h), SXM L is short for St Maarten Landing).The red blocks besides the frames indicate the strength of highlight. The darker, the more highlight.

Ablation Study We demonstrate the necessity of components of our model by comparing the accuracy of ablated versions on the YouTube Dataset, as shown in Table 3.

W/o TG omits the temporal graph, so the features from the spatial graphs are directly fed to the prediction layer.

W/o SG abandons the spatial graphs. As a result, the original features are directly fed into a max-pooling layer to obtain the node features in the temporal graph.

W/o Att takes out the attention weight when aggregating the message, and uses the sum operation as the alternative.

From the results in Table 3, we can find that the overall accuracy of VH-GNN is higher than that of any variants. This demonstrates that our decomposed spatial and temporal graphs are useful, and can capture the object semantics as well as global information. Moreover, the results also validate that the proposed graph operation is important.

Qualitative Analysis

Figure 4 shows some sampled frames from 8 videos in both YouTube and SumMe datasets. The highlight score of each frame is first normalized with min-max normalization and then mapped to color scale. Deeper color beside the frame reflects higher highlight strength. We can easily see that our model can well discriminate the highlight and non-highlight frames. For example, in Figure 4(a), the first 4 frames get higher scores than the 5–8th frames. The reason is that the dog in the 5–8th images is far from the shot and small, thus these frames are unlikely to be highlight.

Conclusion

In this paper, we present VH-GNN for efficient video highlight detection. Our model achieves the state-of-the-art performance and is significantly fast. Specifically, we incorporate decomposed spatial and temporal graphs to capture object semantics and global information, which contributes to the video highlight detection. Although our proposed video highlight detector can achieve favorable results in the two benchmark datasets, our method can still be improved. Firstly, the proposed VH-GNN dose not achieve the topmost performance when handling the indoor scenes with fixed camera and similar objects. Actually, we can introduce the relative-position of objects as a feature to record the motion of objects. Other action/motion features can also be used to improve the proposed method. Secondly, since different videos have different lengths, in this paper, we simply consider one video in a training batch. In fact, we can uniformly sample fixed number of frames from each video, then train the model with better parallel processing capability.

Acknowledgments

This work was supported by National Key Research and Development Program of China (No. 2018AAA0100604, 2017YFB1002804), National Natural Science Foundation of China (No. 61721004, 61872424, 61702511, 61720106006, 61728210, 61751211, 61620106003, 61532009, 61572498, 61572296, 61832002, U1836220, U1705262), the Key Research Program of Frontier Sciences, CAS, Grant NO. QYZDJSSWJSC039, and Youth Innovation Promotion Association CAS 2018166. This work was also supported by Research Program of National Laboratory of Pattern Recognition (No. Z-2018007).

References

Baradel, F.; Neverova, N.; Wolf, C.; Mille, J.; and Mori, G. 2018. Object level visual reasoning in videos. In *ECCV*, 106–122.

Chu, W.-S.; Song, Y.; and Jaimes, A. 2015. Video cosummarization: Video summarization by visual co-occurrence. *CVPR* 3584–3592.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 3844–3852.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. *CVPR* 248–255.

Deng, Z.; Vahdat, A.; Hu, H.; and Mori, G. 2016. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *CVPR*, 4772–4781.

Gao, J.; Zhang, T.; Yang, X.; and Xu, C. 2017. Deep relative tracking. *IEEE TIP* 26(4):1845–1858.

Gao, J.; Zhang, T.; and Xu, C. 2017. A unified personalized video recommendation via dynamic recurrent neural networks. In *MM*, 127–135. ACM.

Gao, J.; Zhang, T.; and Xu, C. 2019a. Graph convolutional tracking. In *CVPR*, 4649–4659.

Gao, J.; Zhang, T.; and Xu, C. 2019b. I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. In *AAAI*, 8303–8311.

Gygli, M.; Grabner, H.; Riemenschneider, H.; and Gool, L. V. 2014. Creating summaries from user videos. In *ECCV*, 505–520.

Gygli, M.; Song, Y.; and Cao, L. 2016. Video2gif: Automatic generation of animated gifs from video. In *CVPR*, 1001–1009.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. B. 2018. Mask r-cnn. *IEEE TPAMI*.

Jain, M.; Van Gemert, J. C.; and Snoek, C. G. 2015. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, 46–55.

Jiao, Y.; Li, Z.; Huang, S.; Yang, X.; Liu, B.; and Zhang, T. 2018. Three-dimensional attention-based deep ranking model for video highlight detection. *IEEE TMM* 20(10):2693–2705.

Kipf, T. N., and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Li, Y.; Zemel, R.; Brockschmidt, M.; and Tarlow, D. 2016. Gated graph sequence neural networks. In *ICLR*, 4453–4462.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 3538–3545.

Lin, T.-Y.; Maire, M.; Belongie, S. J.; Bourdev, L. D.; Girshick, R. B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755.

Ma, C.-Y.; Kadav, A.; Melvin, I.; Kira, Z.; AlRegib, G.; and Peter Graf, H. 2018. Attend and interact: Higher-order object interactions for video understanding. In *CVPR*, 6790–6800.

Marszałek, M.; Laptev, I.; and Schmid, C. 2009. Actions in context. In *CVPR*, 2929–2936.

Mendi, E.; Clemente, H. B.; and Bayrak, C. 2013. Sports video summarization based on motion analysis. *Computers Electrical Engineering* 39:790–796.

Qi, M.; Qin, J.; Li, A.; Wang, Y.; Luo, J.; and Van Gool, L. 2018. stagnet: An attentive semantic rnn for group activity recognition. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *ECCV*, 104–120. Cham: Springer International Publishing.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE TPAMI* 39:1137–1149.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The graph neural network model. *IEEE TNN* 20(1):61–80.

Sun, Y.; Wu, Z.; Wang, X.; Arai, H.; Kinebuchi, T.; and Jiang, Y.-G. 2016. Exploiting objects with lstms for video categorization. In *MM*, 142–146. ACM.

Sun, C.; Shrivastava, A.; Vondrick, C.; Sukthankar, R.; Murphy, K.; and Schmid, C. 2019. Relational action forecasting. In *CVPR*, 273–283.

Sun, M.; Farhadi, A.; and Seitz, S. M. 2014. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 787–802.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 5998–6008.

Wang, X., and Gupta, A. 2018. Videos as space-time region graphs. In *ECCV*, 413–431.

Wang, M.; Fu, W.; Hao, S.; Liu, H.; and Wu, X. 2017. Learning on big graph: Label inference and regularization with anchor hierarchy. *IEEE TKDE* 29:1101–1114.

Wu, Z.; Fu, Y.; Jiang, Y.-G.; and Sigal, L. 2016. Harnessing object and scene semantics for large-scale video understanding. In *CVPR*, 3112–3121.

Wu, J.; Wang, L.; Wang, L.; Guo, J.; and Wu, G. 2019a. Learning actor relation graphs for group activity recognition. In *CVPR*, 9964–9974.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2019b. A comprehensive survey on graph neural networks. *CoRR* abs/1901.00596.

Xiong, B.; Kalantidis, Y.; Ghadiyaram, D.; and Grauman, K. 2019. Less is more: Learning highlight detection from video duration. 1258–1267.

Yang, H.; Wang, B.; Lin, S.; Wipf, D. P.; Guo, M.; and Guo, B. 2015a. Unsupervised extraction of video highlights via robust recurrent auto-encoders. *ICCV* 4633–4641.

Yang, X.; Zhang, T.; Xu, C.; and Hossain, M. S. 2015b. Automatic visual concept learning for social event understanding. *IEEE TMM* 17(3):346–358.

Yang, X.; Zhang, T.; and Xu, C. 2014. Cross-domain feature learning in multimedia. *IEEE TMM* 17(1):64–78.

Yang, X.; Zhang, T.; and Xu, C. 2016. Semantic feature mining for video event understanding. *TOMM* 12(4):55.

Yao, T.; Mei, T.; and Rui, Y. 2016. Highlight detection with pairwise deep ranking for first-person video summarization. In *CVPR*, 982–990.

Zhang, T.; Liu, S.; Xu, C.; Liu, B.; and Yang, M.-H. 2018. Correlation particle filter for visual tracking. *IEEE TIP* 27:2676–2687.

Zhang, Z.; Cui, P.; and Zhu, W. 2018. Deep learning on graphs: A survey. *CoRR* abs/1812.04202.

Zhao, B.; Li, X.; and Lu, X. 2018. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *CVPR*, 7405–7414.

Zhou, B.; Andonian, A.; Oliva, A.; and Torralba, A. 2018a. Temporal relational reasoning in videos. In *ECCV*, 803–818.

Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; and Sun, M. 2018b. Graph neural networks: A review of methods and applications. *CoRR* abs/1812.08434.