# Unsupervised Video Summarization via Deep Reinforcement Learning With Shot-Level Semantics

Ye Yuan and Jiawan Zhang, *Senior Member, IEEE*

***Abstract*— Video summarization is one of the critical techniques in video retrieval, video browsing, and management. It is still a challenging research task due to user subjectivity, excessive redundant information, and lack of spatio-temporal dependency. In this paper, we propose an unsupervised video summarization approach via reinforcement learning with shot-level semantics. The primary idea of this unsupervised method is based on the encoder-decoder model. We use a novel field size dataset to train a convolutional neural network as an encoder to extract the convolutional feature matrix from the video. Then, a bidirectional LSTM is utilized as a decoder to obtain probability weights for selecting keyframes, which preserves the spatio-temporal dependence of video summarization. Specifically, to reduce the influence of user subjectivity, we design a shot-level semantic reward function to generate more representative summarization results. The shot-level semantics are the rules followed by the video shooting process without being changed by the preferences of different viewers. Finally, we evaluate our approach on four classical datasets, SumMe, TVSum, CoSum, and VTW. The results suggest that our algorithm outperforms others and achieves satisfactory results.**

***Index Terms*— Video summarization, deep reinforcement learning, shot-level semantics.**

## I. Introduction

VIDEO summarization, also called video abstract extraction, is necessary for video annotation [1], browsing [2] and retrieval [3]. Video summarization is a selection optimization problem that requires selecting highly representative frames or shots and composing video summaries through a complex mechanism. The clips of the video summary have to contain not only the shortest scenes and events in the video, but also not lose important details. Due to a large number of events, shots, and scenes in the video, it creates a huge challenge for fast automatic extraction of video summaries.

The purpose of the video summary is to provide a brief summary of the video. In general, there are two basic categories: static video abstract and dynamic video skimming. A static abstract, also known as a storyboard, is a collection of highlight images or keyframes extracted from the original video. A dynamic skimming is composed of a few key shots from the original video, but with a much shorter length. Video skimming can support the presentation of motion information and audio data, making the summary more interesting and meaningful for the user. As video skimming and storyboarding can be transformed into each other under certain conditions, we focus on the extraction scheme for the more complex video skimming.

Due to the wide use of video summarization, it has been researched for more than 20 years and many approaches have been proposed, e.g. [4], [5], [6], [7], [8]. During these years, researchers have developed many approaches using feature engineering. These methods provide the ability to extract and select key frames by exploring visual features such as color [9], motion [10], [11], gestures [12], [13], dynamic content [14], objects [15], [16], audio [17], and subtitles [18], etc. The common denominator of these methods is the extraction and comparison of features for each frame, which ignores inter-frame relationships and lacks spatio-temporal dependence. This leads to the selection of redundant keyframes.

With the improved performance of machine learning [19] in dealing with vision problems, many video summarization methods based on machine learning techniques have been proposed. They can be roughly divided into three categories: supervised, unsupervised, and weakly supervised. In the supervised part, Panda *et al.* [20] extracted visual content for video summarization based on CNN architecture. Meanwhile, Zhao *et al.* [21] used LSTM networks to achieve effective video summarization while retaining spatio-temporal dependencies. Several researchers [22], [23] have upgraded and improved this network framework. Some researchers have improved and extended it by introducing attention mechanisms [24] and importance scores [1], among others. However, the supervised learning approaches need to label a large amount of ground-truth data and importance scores, which is a highly time-consuming and tedious task. And the user subjectivity can also cause some limitations in these methods. On the unsupervised side, these methods attempt to learn video summaries without using ground truth data, specifically using Generative Adversarial Networks (GANs). Mahasseni *et al.* [25] designed a GANs framework to train dppLSTM models. Zhou *et al.* [26] used both the diversity of frame levels and the representativeness of the generated summaries as unsupervised training rewards. Besides that, In addition, DR-DSN [26], SUM-GAN-sl [27], SUM-GANrep/

Manuscript received 29 March 2022; revised 6 July 2022; accepted 3 August 2022. Date of publication 10 August 2022; date of current version 6 January 2023. This work was supported in part by the study and demonstration application of organization and service of cultural heritage knowledge graph, in part by the National Key Research and Development Program of China under Grant 2019YFC1521200, and in part by the National Natural Science Foundation of China (NSFC) under Grant 62172295. This article was recommended by Associate Editor J. Shen. *(Corresponding author: Jiawan Zhang.)*

The authors are with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: jwzhang@tju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3197819.

Digital Object Identifier 10.1109/TCSVT.2022.3197819

SUM-GANdpp [25], Cycle-SUM [28], UnpairedVSN [29] and SUM-GAN-AAE [30] are all variants of algorithms that make use of LSTMs or GANs. In terms of weak supervision, researchers have utilized priori factors such as video categories [31], domain knowledge [32], and network images [10], [33] to enhance the performance. However, these priori factors cannot reflect the multimodal features of the video and lead to the loss of critical information.

Although these above approaches have yielded better results in some practical problems, the video summarization is not solved yet. It still suffers from several major drawbacks. First of all, a major problem in feature engineering-based video summarization studies is the lack of interframe temporal dependency. When the shot duration is too long, the spatio-temporal dependency decreases leading to false detection, which is the main reason for the correct result rate below 40%. Secondly, the majority of machine learning-based video summarization methods are limited to the optimization of network frameworks and network structures, which ignore the influence of user subjectivity. Different users have different preferences for summarization, and annotators may have different perspectives. This has caused most methods to yield good results in one dataset and low accuracy in another. Third, to retain the useful information in the original video, small differences in frames are considered keyframes which cause keyframe redundancy. Conversely, there is a loss of critical information. Hence, a general, efficient and stable video summarization method is an urgent research task.

In this paper, we propose an unsupervised deep reinforcement learning method with the shot-level semantic reward for video summarization. The aim of this paper is to reduce the impact of user subjectivity on video summarization by introducing shot-level semantics instead of relying on the visual features of the captured frame, which is inspired by the photography standards followed in video shooting and production. Shot-level semantics refers to the shooting information generated by camera position changes, focus adjustments, and different frame composition styles during the video shooting process. The shot-level semantics are independent of the subject being filmed and include factors such as scene, camera angle, and camera movement, as shown in Figure 1. Unlike the visual features of the captured frames, shot-level semantics do not change due to color, illumination, or object movement, let alone subjective disagreement between different users' understanding of the image. Since shot-level semantics can provide powerful stability and objectivity, we integrate unsupervised deep reinforcement learning methods with it. While preserving inter-frame temporal dependency using unsupervised deep reinforcement learning, the influence of user subjectivity is lifted.

In our approach, the video summarization network is composed of two parts which are the convolutional neural network (CNN) representing the encoder and the bidirectional LSTM network representing the decoder. First, to tackle the challenge of user subjectivity, we use a CNN as an encoder. In contrast to other methods of training convolutional neural networks, we train the network based on a completely new dataset. Our dataset contains 27,000 frames of field size images. In this dataset, each video shot is given special shot-level semantics, such as establishing shot, full shot, close-up shot, etc. In general, CNN training often uses ImageNet datasets, but for tasks such as video summarization, it is difficult to extract suitable feature matrices due to the inconsistent video topics and the high subjectivity of users. However, CNN training with shot-level semantic data will extract as many common feature matrices as possible to avoid the influence of user preferences on the summarization results thereby improving the quality of the results. Then, a bidirectional LSTM network is used as a decoder to yield probability weights for keyframe selection, thus keeping the spatio-temporal dependence. Finally, based on reinforcement learning, we creatively design rules for reward functions that utilize shot-level semantic rules instead of inter-frame similarity to produce more diverse and representative summaries. Specifically, the reward function consists of shot duration, field size, and motion information features. The field size reward reflects the diversity that can be measured for each frame. On the other hand, the motion reward is calculated by the motion feature of each shot to guide the acquisition of more representative keyframes. Hence, this reward function is designed to be more consistent with the human habit of video summarization. The qualitative results show that our algorithm yields significant improvements in precision and recall.

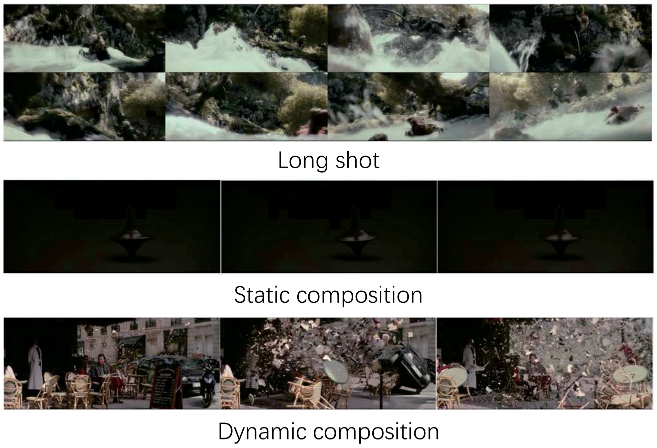


Fig. 1. Examples of different shot-level semantics.

In particular, the main contributions of this paper in the video summarization are as follows.

- We propose a deep reinforcement learning unsupervised video summarization algorithm whose performance reaches state-of-the-art mainly due to the introduction of shot-level semantics instead of traditional visual features. To the best of our knowledge, this paper is the first study to utilize shot-level semantics in an unsupervised video summarization approach.
- We create a new large shot-level semantic dataset for training CNN networks to address the problem of user subjectivity. Moreover, we will release the dataset to encourage future research. A reward function based on shot-level semantics is designed to improve the representativeness and diversity in video frame selection.

- We have conducted experiments on four standard datasets. The experimental results show that our algorithm is not only the most advanced among unsupervised algorithms, but also not inferior to supervised methods.

The rest of the paper is organized as follows. Section II describes a brief survey on related work. Section III describes the unsupervised video summarization algorithm with shot-level semantic reward based on reinforcement learning. The experiments and evaluations are provided in Section IV. Finally, we give the conclusions and discuss possible future work in Section V.

## II. Related Work

Video summarization is a highly active research topic in multimedia analysis, with rich literature resources [34]. There are three important steps in video summarization including, video preprocessing, selection of meaningful frames representing the video content, and synthesis of the output results. The difference between these algorithms is the model built when selecting the representative frames. It can be divided into two types: feature engineering based, and deep learning based models. Among them, the deep learning based methods can be further classified into supervised video summarization, unsupervised video summarization, and weakly supervised video summarization. In this section, we briefly review the relative work of video summarization in all these areas.

### *A. Feature Engineering-Based Video Summarization*

Video summarization methods based on feature engineering are more commonly found in the literature. Researchers [35] summarized the video by using the color histogram. Zhang *et al.* [36] also described this technique, in which the keyframe selection is carried out based on texture and color attributes. While color features are also a useful aspect of video analysis, most color histogram methods are used in shot detection. Besides color features, Li *et al.* [37] present a discussion of techniques working on constant velocity motion and relative motion. Bulut *et al.* [38] and Ajmal *et al.* [39] both use motion trajectories to analyze the connection between human motion and video summarization. Damnjanovic *et al.* [40] summarized the video according to the detected events by using spectral clustering algorithms. Motion-based methods are beneficial only for videos with moderate motion but limited for videos containing huge motion or service motion. Likewise, color-based techniques are simple but have low accuracy rates. Ali Javed *et al.* [41] proposed a keyframe extraction summary by identifying audio content. It is generally considered that single visual, trajectory, audio, and other features are used for keyframe selection on the basis of feature engineering, which is not far enough for a proper summary generation. Moreover, these methods mostly examine only visual clues with no regard to the sequence structure of the video and the temporal dependencies between frames.

### *B. Deep Learning-Based Model*

With the development of machine learning algorithms, there is an increasing number of advanced deep neural network architectures used to learn video summarization. We provide a comprehensive survey of existing deep learning-based video summarization methods with a description of supervised, unsupervised, and weakly supervised video summarization approaches.

*1) Supervised Video Summarization:* Supervised video summarization learns importance scores by modeling the temporal dependencies between frames. Zhang *et al.* [42] were the first to propose the use of long short-term memory (LSTM) units [43] to model the variable range temporal dependence between video frames. The importance of frames was analyzed by a multilayer perceptron (MLP) and the decision point process (DPP) [44]. Then, Zhao *et al.* [21] introduced a two-layer LSTM architecture. Lebron Casas *et al.* [23] built on this by introducing an attention mechanism to model the evolution of user interests over time. Fajtl *et al.* [24], on the other hand, use the attention mechanism as a core part of the analysis, aiming to avoid the limitations of the LSTM. Liu *et al.* [45] describe a hierarchical approach to estimate the representativeness of each shot and define a set of candidate keyframes. Ji *et al.* [1] generalized video summarization to the seq2seq learning problem and extended this model [46].

Not only that, Rochan *et al.* [47] utilized semantic models like Fully Convolutional Networks (FCN) [48] and DeepLab [49] to analyze video summarization. Lal *et al.* [50] modeled spatio-temporal relationships in video through an encoder-decoder architecture with a convolutional LSTM. Based on this architecture, several researchers [51], [52] have coincidentally thought of using CNNs to extract features from video content and input these into LSTMs thus modeling the spatial and temporal structure of the video. Huang *et al.* [6] propose a transition effect detection (TED) method to improve shot segmentation accuracy and complete comprehensive video summarization by motion information. Paul *et al.* [53] propose a new video summarization framework based on eye tracker data, which uses the distance between the viewer's current and previous focus points to calculate motion salience scores for video summarization. Chu *et al.* [54] added optical flow algorithms to enhance the accuracy on top of processing the original frames using CNNs. Although supervised video summarization methods can yield relatively promising results, they are overly relying on manually labeled ground truth data, which is quite expensive, difficult, and time-consuming to obtain, and in some aspects, it even becomes impossible [29], [55].

*2) Unsupervised Video Summarization:* Unsupervised video summarization can learn by fooling the discriminator. The unsupervised approach lacks guidance with ground truth data so it relies on human determined rules. Representative summaries should help focus on inferring the content of the original video. In this case, some techniques use LSTM to learn how to create video summaries. Most algorithms optimize the LSTM by adding attention mechanisms in order to obtain a reconstructed summary video. Some techniques use GANs to learn to create video summaries in an adversarial manner where the summarizer tries to trick the discriminator to get a reconstructed summary video. Some unsupervised methods use reinforcement learning principles to improve the quality of the results by incorporating a reward function to quantify the expected features present in the generated summaries.

Mahasseni *et al.* [25] use an LSTM-based frame selector to learn video summarization through an adversarial learning process. This algorithm aims to minimize the distance between the original video and the reconstructed version. Apostolidis *et al.* [27] improved this method and proposed a label-based approach to train the adversarial part of the network, which improves the summarization performance. Yuan *et al.* [28] utilize trainable several discriminators and a trainable loop with consistent adversarial learning objective to maximize the information between summaries and videos. Shi *et al.* [56] use attribute mining and inference to associate different attributes with global appearance features and discover their potential relationships to generate a more comprehensive description. Not only that, Tokmakov *et al.* [57] introduced classinstance recognition and local aggregation to unsupervised video summarization for capturing motion patterns in videos thus enhancing the summarization algorithm. Ma *et al.* [4] proposed a similarity based block sparse subset selection (SB2S3) model that uses inter-frame similarity to consider global relationships and local relationships through sparsity to obtain video summarization results.

Not only that, Apostolidis *et al.* [5] embed the actor-critic model into GAN to enhance the applicability of the model. He *et al.* [58] proposed a conditional GAN based on the self-attention model. In addition, video summaries can also be learned by summarizing the rules for user expectation summaries. In this case, researchers have used reinforcement learning combined with a reward function to quantify the expectations of video summaries. Zhou *et al.* [26] proposed diversity-representative rewards to train summarized video summaries. Yaliniz *et al.* [59] enriched the reward function and considered the uniformity of the generated summaries. Gonuguntla *et al.* [60] use a spatio-temporal segment network to extract spatial and temporal information of video frames and summarize the video by a reward function. The unsupervised video mainly relies on understanding the subjectivity of users and generalizing the summary rules. However, different users have different preferences for summaries with variable importance rules. Therefore it is difficult to generalize consistent significance reward rules. Inspired by video shooting, we use shot-level semantic features to extract as much as possible a uniformity reward function that matches the user's browsing.

*3) Weakly Supervised Video Summarization:* Weakly supervised video summarization methods also attempt to reduce the requirement for large amounts of ground truth data. Cai *et al.* [61] proposed a weakly supervised setup for learning summarization from a large number of web videos. Ho *et al.* [62] presented a deep learning framework for first-person videos. Chen *et al.* [63] used the principles of reinforcement learning to build and train a summarization model based on limited human annotations. Weak supervision often utilizes priori factors such as interestingness [64] and gaze [65], video categories [31], domain knowledge [32], and network images [10], [33] to facilitate the summarization process. Even though the need for real data is reduced, these a priori factors cannot reflect the specific content of the videos.

## III. Our Approach

In this section, we will describe the details of our algorithm and the computational process. Figure 2 shows the pipeline of our unsupervised video summarization algorithm. We propose an approach based on deep reinforcement learning with shot-level semantics to efficiently extract important keyframes in the video. Our algorithm efficiently trains the video in an end-to-end mode and encodes the high-level semantic information of the video using shot-level semantic features. Therefore, it can handle the diverse and complex scenes in the video pretty well. First, we use CNN to treat the frames in the video and get the convolutional feature matrix. The training data used for the convolutional neural network is containing 27,000 images of camera scenes. After that, we sent the convolutional feature matrix into a bidirectional LSTM network to determine which keyframes to choose based on the predicted probability distribution. Finally, we design a reward function based on shot-level semantics that incorporates field size and shot motion features for assessing the weight of the selected frames. In particular, the inclusion of shot-level semantics in the algorithm can effectively avoid user subjectivity and make the results more representative. The algorithm is described in more detail as follows.

### A. Definition of Shot-Level Semantics

Shot-level semantics include camera shooting mode, frame composition, field sizes. The same shot-level semantics give the viewer the same feeling. For example, an establishing shot tends to have a heavy, serene mood. Dynamic compositions can give a tense atmosphere, etc. Shot-level semantics include the following categories. Camera shooting mode mainly includes long shots and short shots. The length of the shot is utilized to judge whether it is a long shot or a short shot. In general, a shot with a duration of more than 10 seconds is called a long shot as in Figure 1, on the contrary, it is called a short shot. Frame composition contains static composition and dynamic composition, as in Figure 1. The static composition represents a relatively stationary object or moving object temporarily in a static state, while dynamic composition shows that the object is constantly changing. Field sizes refer to the difference in the range of the subject presented in the frame, which is generated by the different distances between the subject and the camera. The field size includes establishing shot, master shot, wide shot, full shot, medium full shot, medium shot, medium close up, close up, and extreme close up, as shown in Figure 3. The above shot-level semantics are summarized based on shooting techniques. Our goal is to reduce the impact of user subjectivity on video summarization by using shot-level semantics instead of visual features in the unsupervised video summarization process.

### B. Convolutional Neural Network Based Encoder

As discussed in the literature, video summarization needs to predict and make optimal choices for the importance scores of frames. Before starting, we preprocess the video, which can reduce the computational complexity and improve efficiency.

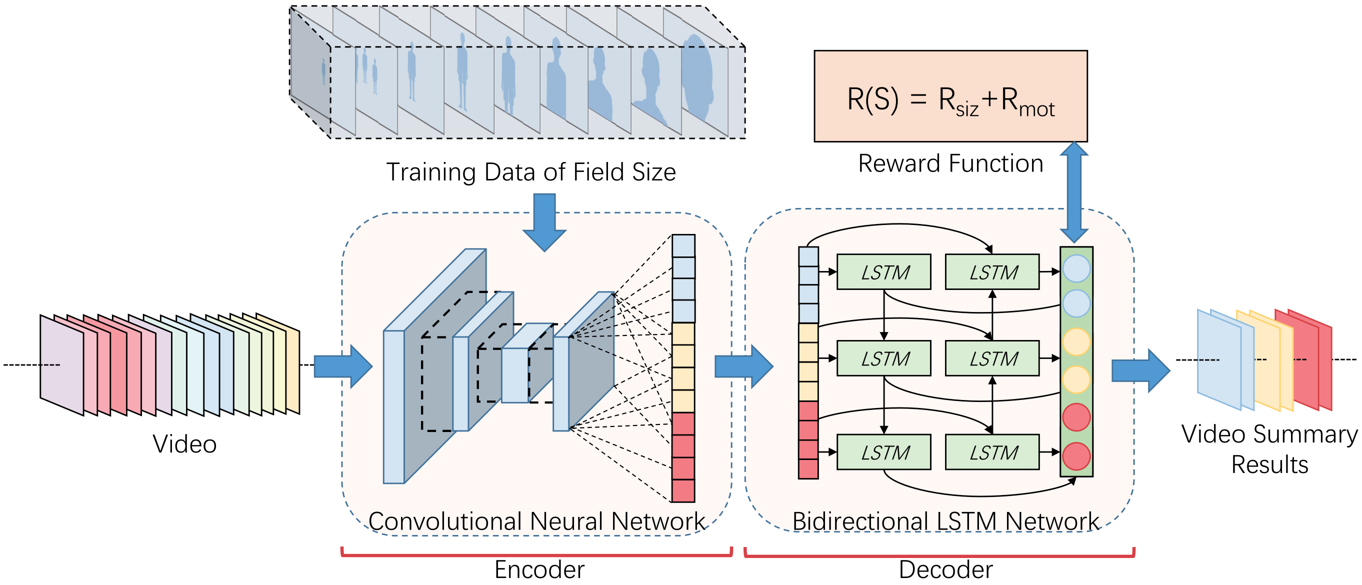


Fig. 2. The video summarization pipeline based on SL-DSN with encoder and decoder to extract the video highlights.

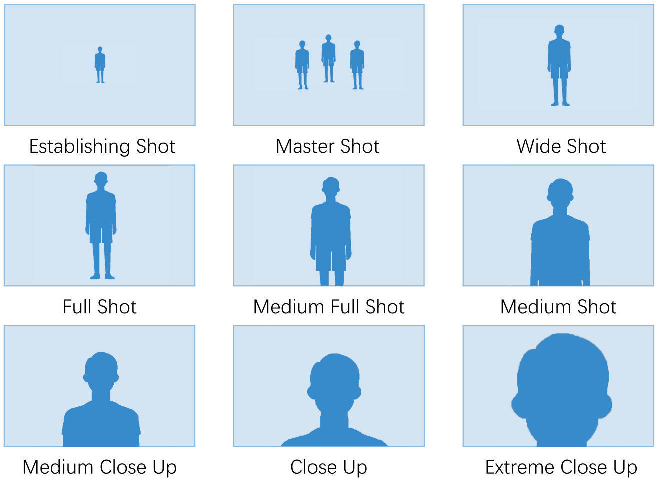


Fig. 3. The nine different categories of field sizes.

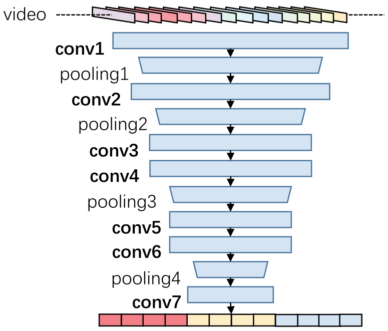


Fig. 4. The architecture of CNN.

We segment the video using the shot detection algorithm proposed by Yuan *et al.* [66] and obtain one frame from each shot as a representative. This algorithm uses dynamic mode decomposition to decompose a video into several shots. Since the content within a shot represents the same storyline, annotators tend to give importance scores on a shot-by-shot basis. In other words, the importance scores within the same shot are coherent. We denote the shots extracted from the dynamic mode decomposition as $S_T^1 = [s_1, s_2, s_3, \ldots, s_T]$, where $S_i(1łiłT)$ represents the $ith$ video shot in the video. The collection of representative shot frames extracted by preprocessing is given as $X_T^1 = [X_1, X_2, X_3, \ldots, X_T]$, where $X_i$ stands for the frame of $S_i$ shot.

Subsequently, each frame is sent into the CNN network to obtain the convolutional feature matrix. Inspired by the video shooting style, we use field size instead of the ImageNet dataset [67] to complete the network training. The shot view is a composition mode usually used in video shooting, as shown in Figure 3. The field size includes establishing shot, master shot, wide shot, full shot, medium full shot, medium shot, medium close up, close up, and extreme close up. The rationality of this idea comes from the observation of the video since a good video summarization will satisfy the following two laws. 1) In the video, the duration of the clip that is too long will not be selected by the user in the video summary. 2) Telephoto shots are often narrating too much detail that will not be helpful to the overview of the video from the user.

Moreover, the features in these two laws do not cause great volatility with user subjectivity. Accordingly, we created a new dataset containing multiple field size images and semi-automatically generated tags.

Figure 4 shows the structure of the CNN model that we used. We take each frame from each shot in the video $X_T^1 = [X_1, X_2, X_3, \ldots, X_T]$ as input into the encoder to capture the shot-level semantic features. We used a CNN network model containing seven convolutional layers and four maxpooling layers. As shown in Table I, the convolution kernels of the top six convolutional layers are all 3*3 with one stride and padding size of 1. In particular, we added BatchNormalization after conv5 and conv6 to accelerate the convergence of the model while reducing the training time

TABLE I

NETWORK CONFIGURATION SUMMARY. k, s AND p REPRESENT THE KERNEL SIZE, STRIDE, AND PADDING SIZE

| Type | Configurations |
|---|---|
| Input | W*64 gray-scale frames |
| Convolution1 | #map: 64, k: 3*3, s: 1, p: 1 |
| MaxPooling1 | Window: 2*2, s:2 |
| Convolution2 | #map: 128, k: 3*3, s: 1, p: 1 |
| MaxPooling2 | Window: 2*2, s:2 |
| Convolution3 | #map: 256, k: 3*3, s: 1, p: 1 |
| Convolution4 | #map: 256, k: 3*3, s: 1, p: 1 |
| MaxPooling3 | Window: 1*2, s:2 |
| Convolution5 | #map: 512, k: 3*3, s: 1, p: 1 |
| BatchNormalization | - |
| Convolution6 | #map: 512, k: 3*3, s: 1, p: 1 |
| BatchNormalization | - |
| MaxPooling4 | Window: 1*2, s:2 |
| Convolution7 | #map: 1024, k: 4*4, s: 1, p: 0 |

greatly. There are four maxpooling layers in the network which we add after the conv1, conv2, conv4, and conv6 layers. However, the window size of the last two maxpooling layers is adjusted from 2*2 to 1*2. The purpose of this is to not lose the width direction information as much as possible and to be more consistent with the recognition of the field size. With the last convolutional layer conv7, which is composed of 4*4 convolutional kernels with one stride and a padding size of 0, we will get 1024 neurons as the convolutional feature matrix into the decoder.

### *C. Bidirectional LSTM Based Decoder*

Bidirectional LSTM is a special recurrent neural network. We choose it because it can preserve the spatio-temporal dependence while using the bidirectional propagation mechanism can greatly reduce the loss of information in the context, which is very effective in video analysis tasks. Besides, the BiLSTM does not require complicated hyperparameter debugging and effectively alleviates the problem of exploding gradients that traditional RNN models tend to produce. The principle of BiLSTM is to split the neurons of normal LSTM into two directions, i.e., forward time direction and backward time direction. In particular, the outputs of two hidden states are not connected. The past and future information of the current frame can be saved by using the status in both directions.

Figure 5 shows the BiLSTM network structure. The forward state order reads the feature matrix $x_t$ extracted by the CNN and computes the forward hidden state $\overrightarrow{h}_t$. Meanwhile, the backward state is read in decreasing order from the feature matrix $x_t$ extracted by the CNN and computed backward to the hidden state $\overleftarrow{h}_t$. The output gate $o_t$ corresponding to each input $x_t$ is gained by computing the forward and the backward hidden states. In other words, each output gate $o_t$ carries the shot-level semantic information of the before and after frames.

The convolutional feature matrix of the fully connected layer from the CNN is put into the BiLSTM. The core of the LSTM is the memory unit, which serves to encode all the knowledge and inputs up to that step. the most important in the LSTM are the input gate, the forget gate, and the output gate. The input gate $i_t$ considers how much of the network input $x_t$ is saved to the cell state in the current time, which is calculated as follows.

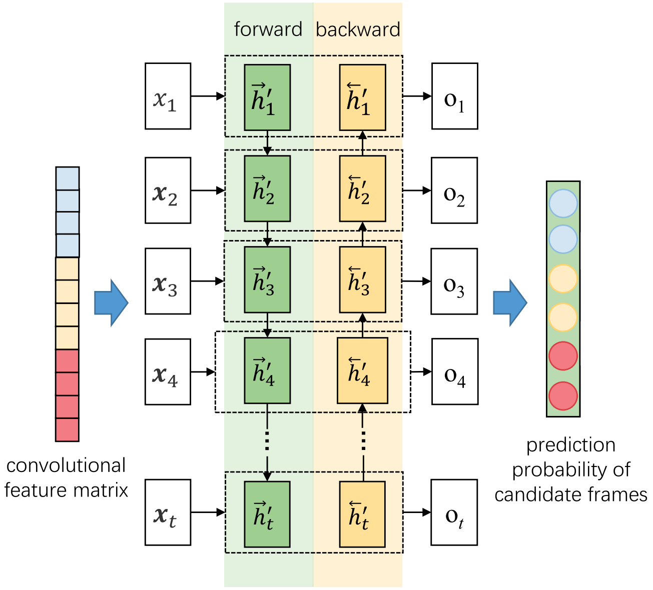


Fig. 5. The architecture of BiLSTM.

$$i_t = sigmoid(W_i[x_t^T, h_{t-1}^T]^T) \tag{1}$$

The forget gate $f_t$, on the other hand, allows forgetting of the previous memory $c_t$, i.e.

$$f_t = sigmoid(W_f[x_t^T, h_{t-1}^T]^T) \tag{2}$$

$$c_t = i_t \odot tanh(W_c[x_t^T, h_{t-1}^T]^T) + f_t \odot c_{t-1} \tag{3}$$

The output gate $o_t$ determines how much memory can be transferred to the hidden state $h_t$.

$$o_t = sigmoid(W_o[x_t^T, h_{t-1}^T]^T) \tag{4}$$

where $W_i, W_f, W_c, W_o$ are the training weights and $sigmoid(\cdot)$ is the activation function. With the predicted probability of the output, the summary candidate frames are generated using Bernoulli sampling as follows,

$$\alpha_t = Bernoulli(o_t) \tag{5}$$

$_t \in (0, 1)$ indicates whether the t$th$ shot is selected for the abstract. In addition, when training summaries in a supervised environment with human annotations, we use the Mean Square Error to measure the loss of prediction, which is calculated as follows.

$$l^p = \frac{1}{m} \|o - g\|_2^2 \tag{6}$$

where $o$ and $g$ denote the vector predicted by our method and the vector annotated by humans. $\|\cdot\|$ is the 2-Norm.

### *D. Reward Functions Based on Shot-Level Semantics*

In reinforcement learning, we evaluate the summaries by means of a reward function. The process of maximizing the reward function is the step of generating high-quality summaries. Inspired by video shooting, we construct the reward function using shot-level semantics, including shot duration, shot scene, and shot motion. While ensuring image features, we avoid as much as possible the influence due to user subjectivity.

We evaluate the diversity of the summaries through the duration of the shots in which the selected frames are located. In general, based on our observations of human-annotated video summarization, clips that are too long or too short in duration are often not selected for inclusion in video summarization, because they either have a great deal of redundant information or have little useful information owing to their short duration. For this reason, we propose a shot duration reward. We denote the shot duration as $D_T^1 = [d_1, d_2, d_3, \ldots, d_T]$, which are calculated as follows.

$$R_{dur} = \frac{1}{|d_t - d_{median} + b|} \tag{7}$$

where $d_median$ represents the median of all shot duration, and $b$ is residual to avoid a divisor of 0, default is 1. When the clip duration is too long or too short less reward will be given. Only the suitable duration would give a higher reward.

The field size reward is to extract more representative keyframes under the same duration reward. In Figure 3, we can understand that shots can be divided into 9 categories, and the commonly used views during video shooting are full shot, medium full shot, and medium shot. For the summary, close-up or establishing shots are the ones based on representativeness, since they contain more shot-level semantic information. We use Tanimoto coefficients to calculate the distance between objects of different sizes

$$R_{fs} = \frac{x_t \cdot x_t'}{\|x_t\|^2 + \left\|x_t'\right\|^2 - x_t \cdot x_t'} \tag{8}$$

It is possible to avoid the lack of information brought about by the direct measurement of screen relevance and to better enhance representative shots.

The shot motion reward can be regarded as the study of the motion of the subject. In video shooting, if the object is relatively static it is called static composition, and otherwise, it is called dynamic composition. For video summaries, motion clips with too much speed are not easily remembered, so the majority of dynamic compositions are discarded in human annotation results. We use YOLOv4 [68] to identify and track objects and calculate the shot motion reward formula as follows

$$R_{mov} = e^{-\left(\sum_{i=1}^{T} \sum_{j-1}^{V} \frac{p_i}{s_i} + b\right)^2} \tag{9}$$

where $s_i$ represents the duration in the $ith$ shot and $p_j$ denotes the pixel distance of the $jth$ object moving in that shot. $V$ represents the total number of moving objects in that shot, $b$ is the motion residual.

We combine the three reward options and define the new shot-level semantics-based reward function as

$$R = R_{dur} + R_{fs} + R_{mov} \tag{10}$$

In the experimental section, it can be learned that our reward function provides strong robustness and plays an important role during the training period.

We optimize the proposed reinforcement learning method using the REINFORCE algorithm [69], by introducing the Monte-Carlo policy gradient method

$$\triangledown J(\theta) = \frac{1}{n} \sum_{n=1}^{N} \sum_{t=1}^{T} (R_n - b) \triangledown_\theta \, log \pi_\theta (\alpha_t \mid h_t) \tag{11}$$

where $\theta$ denotes the trainable parameters of the summarized network, $\alpha_t$ is the Bernoulli sampling candidate frame, and $h_t$ is the hidden state of the BiLSTM. n represents the n$th$ clips, and $b$ is defined as the average of reward $R$. The details of the associated gradient algorithm are not repeated in this paper.

## IV. Experiments

In this section, we use quantitative and qualitative experiments to demonstrate the effectiveness and generality of our algorithm. We give a brief introduction to the dataset and the experimental setup. Then, the proposed method is evaluated in the dataset and compared with existing methods. Not only that, we design qualitative experiments for a deeper analysis of our approach.

### A. Environmental Settings

1) Dataset: We evaluated our algorithm on four public datasets SumMe [64], TVSum [14], CoSum [70], and VTW [71]. The SumMe dataset contains 25 videos, with the duration of the videos ranging from 1 minute to 7 minutes. These videos contain several types, such as outdoor, sports, landscape, etc. SumMe was annotated by 15 to 18 people who were assigned an importance score for each frame. The TVSum dataset is a collection of 50 videos that were downloaded and edited from YouTube, with video durations ranging from 2 to 10 minutes. The videos were divided into 10 categories, that is, changing vehicle tires, grooming an animal, parade, dog show, and so on. TVSum is annotated by 20 people and given an importance score by dividing the videos into sub-clips. Both datasets have a wide variety of video content and include first-person view or third-person view. The third dataset is CoSum which contains 51 videos. This dataset was used for the video co-summarization task. The videos in the dataset are composed of thematic keywords. The fourth dataset is VTW, which is composed of 2529 videos with an average duration of 1 to 2 minutes. These videos were also downloaded from YouTube and manually annotated with data.

2) Evaluation metrics: The most commonly used quantitative evaluation metrics [25], [42] are precision (P), recall (R), and F-score (F), which were defined as

$$P = \frac{S \cap G}{S}, \quad P = \frac{S \cap G}{G}, \quad F = \frac{2 \times P \times R}{P + R} \tag{12}$$

where S represents the summarized video generated by the algorithm and G represents the ground truth video clips. The F-score indicates the overall performance of the algorithm, combining precision and recall. In the process of comparison, we bold the best performing algorithm to distinguish it from other algorithms.

TABLE II

THE PRECISION, RECALL AND F1-SCORE ACHIEVED BY DIFFERENT VARIANTS OF THE PROPOSED METHOD ON THE SUMME AND TVSUM DATASETS

| | SumMe | | | TVSum | | |
|---|---|---|---|---|---|---|
| Method | F1-score | Precision | Recall | F1-score | Precision | Recall |
| $SUM-SL_{dur}$ | 50.6 | 51.9 | 49.8 | 59.0 | 59.1 | 58.9 |
| $SUM-SL_{dur+fs}$ | 51.4 | 52.6 | 50.6 | 60.0 | 59.6 | 60.4 |
| $SUM-ISL_{dur+fs+mov}$ | 50.7 | 50.4 | 51.1 | 58.9 | 58.3 | 59.7 |
| $SUM-SL_{dur+fs+mov}$ | **52.0** | **53.3** | **51.2** | **62.2** | **61.9** | **62.6** |

TABLE III

COMPARISON WITH UNSUPERVISED METHODS ON SUMME AND TVSUM DATASETS

| | | SumMe | | TVSum | | |
|---|---|---|---|---|---|---|
| | Algorithms | F1-score | Rnk | F1-score | Rnk | Avg Rnk |
| CNN+LSTM | Online Motion-AE [72] | 37.7 | 13 | 51.5 | 13 | 13 |
| | CSNet [73] | 51.3 | 3 | 58.8 | 4 | 3.5 |
| LSTM+Attention Mechanism | SUM-FCN [47] | 41.5 | 11 | 52.7 | 12 | 11.5 |
| | CRSUM [51] | 47.3 | 7 | 58.0 | 8 | 7.5 |
| | SUM-GDA [74] | 50.0 | 4 | 59.6 | 3 | 3.5 |
| Reinforcement Learning+Reward Function | DR-DSN [26] | 41.4 | 12 | 57.6 | 9 | 10.5 |
| | EDSN [60] | 42.6 | 10 | 57.3 | 10 | 10 |
| | SUM-Ind [59] | 51.4 | 2 | 61.5 | 2 | 2 |
| GANs | PCDL [75] | 42.7 | 9 | 58.4 | 6 | 7.5 |
| | ACGAN [58] | 46.0 | 8 | 58.5 | 5 | 6.5 |
| | SUM-GAN-AAE [30] | 48.9 | 5 | 58.3 | 7 | 6 |
| | UnpairedVSN [29] | 47.5 | 6 | 55.6 | 11 | 8.5 |
| | Ours | **52.0** | **1** | **62.2** | **1** | **1** |

### *B. Implementation Details*

We use SumMe, TVsum, CoSum, and VTW to conduct our experiments. Specifically, our dataset contains a total of 2655 videos. In our work, the training and test sets are split according to 80% and 20%, i.e., 2100 videos are used for training and 555 videos are used for testing. SumMe, CoSum, and VTW have shot boundary annotation information. For these three datasets, we extracted one frame from the boundaries of the video clips as training samples. However, the TVsum dataset does not have shot boundary annotation information. We segment the video using the shot boundary detection algorithm proposed by Yuan *et al.* [66] and extract a frame at the boundary of each clip for the experiment.

To obtain the features of the video clips, we train the data using a CNN network. We use the last layer of the convolutional layer to get a feature matrix with 1024 neurons. We use 30% of the data in the training dataset with a total of 630 videos as the validation set to determine the hyperparameters. In this paper, we set the dimensionality of the hidden states in the BiLSTM to 256 and the learning rate to 2e-5. The residual b in Eq.7 is set to 1. The motion offset in Eq.9 is set to 3. The number of clips in Eq.11 is set to 10. Training will be stopped when it reaches the maximum number of epochs (120 in our example). It will be possible to stop early when the reward obtained by the reward function increases over a while.

### *C. Quantitative Evaluation*

We designed the ablation experiment for the proposed method and named the method variants. $SUM-SL_{dur}$ means that the reward function used contains only $R_dur$. $SUM-SL_{dur+fs}$ means that the reward function contains $R_dur$ and $R_fs$. $SUM-SL_{dur+fs+mov}$ denotes the version of the reward function that uses the three shot-level semantics of shot duration reward, field size reward, and motion reward. Table II shows the comparison between different variants of our method in two different datasets. The results show that $SUM-SL_{dur+fs+mov}$ has higher precision, recall, and F1 score than the other two variant versions. Using multiple shot-level semantics can better train the network to yield higher quality video summarization results. On the SumMe dataset, $SUM-SL_{dur+fs+mov}$ outperforms the others by 0.6% 1.4%, while on the TVSum dataset, the results are improved by 1.7% 2.7%. These results indicate that utilizing the rewards of shot-level semantics can effectively improve the performance of the algorithm. Likewise, we calculated $SUM-ISL_{dur+fs+mov}$ using ImageNet instead of the scene-level training dataset that we created, and the results have an obvious gap with the $SUM-SL_{dur+fs+mov}$.

We compared with 12 unsupervised methods on two datasets, SumMe and TVSum, including: Online Motion-AE [72], SUM-FCN [47], DR-DSN [26], EDSN [60], UnpairedVSN [29], PCDL [75], ACGAN [58], SUM-GAN-AAE [30], CSNet [73], CRSUM [51], SUM-GDA [74], and SUM-Ind [59]. Table III shows that the proposed algorithm is much better than all other unsupervised methods a maximum of 14.3%. Our algorithm is the highest in terms of F1 score, both for the SumMe and TVSum datasets. Compared to the more classical algorithms Online Motion-AE [72] and CSNet [73], which combine a convolutional neural network with a short-term memory network, these perform 14.3% and 0.7% better on the SumMe dataset. The performance on the TVSum dataset is 10.7% and 3.4% higher. In contrast to the LSTM algorithms with attention mechanism, our algorithm improves 10.5%, 7.4%, and 2% over SUM-FCN [47], CRSUM [51], and SUM-GDA [74], and 9.5%, 4.2%, and 2.6% on the TVSum dataset. With DR-DSN [26], EDSN [60], and SUM-Ind [59] which also use the concept of reward function improved by 10.6%, 9.4%, and 0.6% on the SumMe dataset, and by 4.6%, 4.9%, and 0.7% on the TVSum dataset. Compared to the algorithms PCDL [75], UnpairedVSN [29], ACGAN [58], and SUM-GAN-AAE [30] for GANs, the performance is 9.3%, 6%, 3.1%, and 4.5% higher on the

TABLE IV

COMPARISON WITH SUPERVISED METHODS ON SUMME AND TVSUM DATASETS

| | | SumMe | | TVSum | | |
|---|---|---|---|---|---|---|
| | Algorithms | F1-score | Rnk | F1-score | Rnk | Avg Rnk |
| LSTM | HSA-RNN [22] | 44.1 | 10 | 58.5 | 6 | 8 |
| | SMLD [54] | 47.6 | 5 | 59.6 | 4 | 4.5 |
| LSTM+Attention Mechanism | SF-CVS [42] | 46.0 | 8 | 57.3 | 11 | 9.5 |
| | A-AVS [1] | 43.9 | 11 | 58.4 | 7 | 9 |
| | SUM-FCN [47] | 47.5 | 6 | 57.6 | 10 | 8 |
| | H-MAN [45] | 51.8 | 2 | 61.5 | 2 | 2 |
| | SUM-DeepLab [47] | 48.8 | 4 | 58.8 | 5 | 4.5 |
| | DASP [46] | 45.5 | 9 | 58.0 | 9 | 9 |
| | VASNet [24] | 49.7 | 3 | 61.4 | 3 | 3 |
| GANs | ACGANsup [58] | 47.2 | 7 | 58.3 | 8 | 7.5 |
| | Ours | **52.0** | **1** | **62.2** | **1** | **1** |

TABLE V

COMPARISON METHODS ON COSUM AND VTW DATASETS

| | CoSum | | VTW | |
|---|---|---|---|---|
| Algorithms | F1-score | Rnk | F1-score | Rnk |
| VSUMM [76] | 41.2 | 8 | - | - |
| LiveLight [77] | 51.1 | 7 | - | - |
| Summary Transfer [78] | 65.3 | 5 | - | - |
| vsLSTM [42] | 64.4 | 6 | 44.1 | 5 |
| dppLSTM [42] | 65.5 | 4 | 44.3 | 4 |
| Hierarchical RNN [21] | 66.3 | 3 | 46.5 | 3 |
| HSA-RNN [22] | 69.2 | 2 | 49.1 | 2 |
| Ours | **70.3** | **1** | **49.7** | **1** |

SumMe dataset. For TVSum dataset the performance is higher by 3.8%, 3.7%, 3.9% and 6.6%. In particular, there is still a 0.6% and 0.7% improvement in our algorithm compared to the SUM-Ind [59] algorithm that performs best on both datasets. The result outperforming other algorithms is credited to the fact that our reward function incorporates lens-level semantic features, which can better solve the user subjectivity problem.

In addition, we compare our method with existing supervised methods as shown in Table IV, including SUM-FCN [47], SF-CVS [6], A-AVS [1], HSA-RNN [22], ACGAN-sup [58], SUM-DeepLab [47], DASP [46], SMLD [54], H-MAN [45], and VASNet [24]. In the comparison, we can notice that our algorithm when using the full shot-level semantics, our algorithm shows an advantage for the H-MAN [45] and VASNet [24] algorithms. Our method is not only competitive in results, but also can greatly reduce the reliance on annotated data as it does not require manual annotation.

Besides, we evaluated on two important and commonly used datasets, SumMe and TVSum. To demonstrate the generality of our algorithm, experiments were also performed on two datasets, CoSum and VTW. As shown in Table V, the results from our comparisons with VSUMM [76], LiveLight [77], Summary Transfer [78], vsLSTM [42], dppLSTM [42], Hierarchical RNN [21], and HSA-RNN [22] show that both for the CoSum dataset and VTW dataset, our algorithm performs better than the others. Experiments on the four datasets reveal that introducing shot-level semantics in unsupervised video summarization can effectively improve the quality of the results.

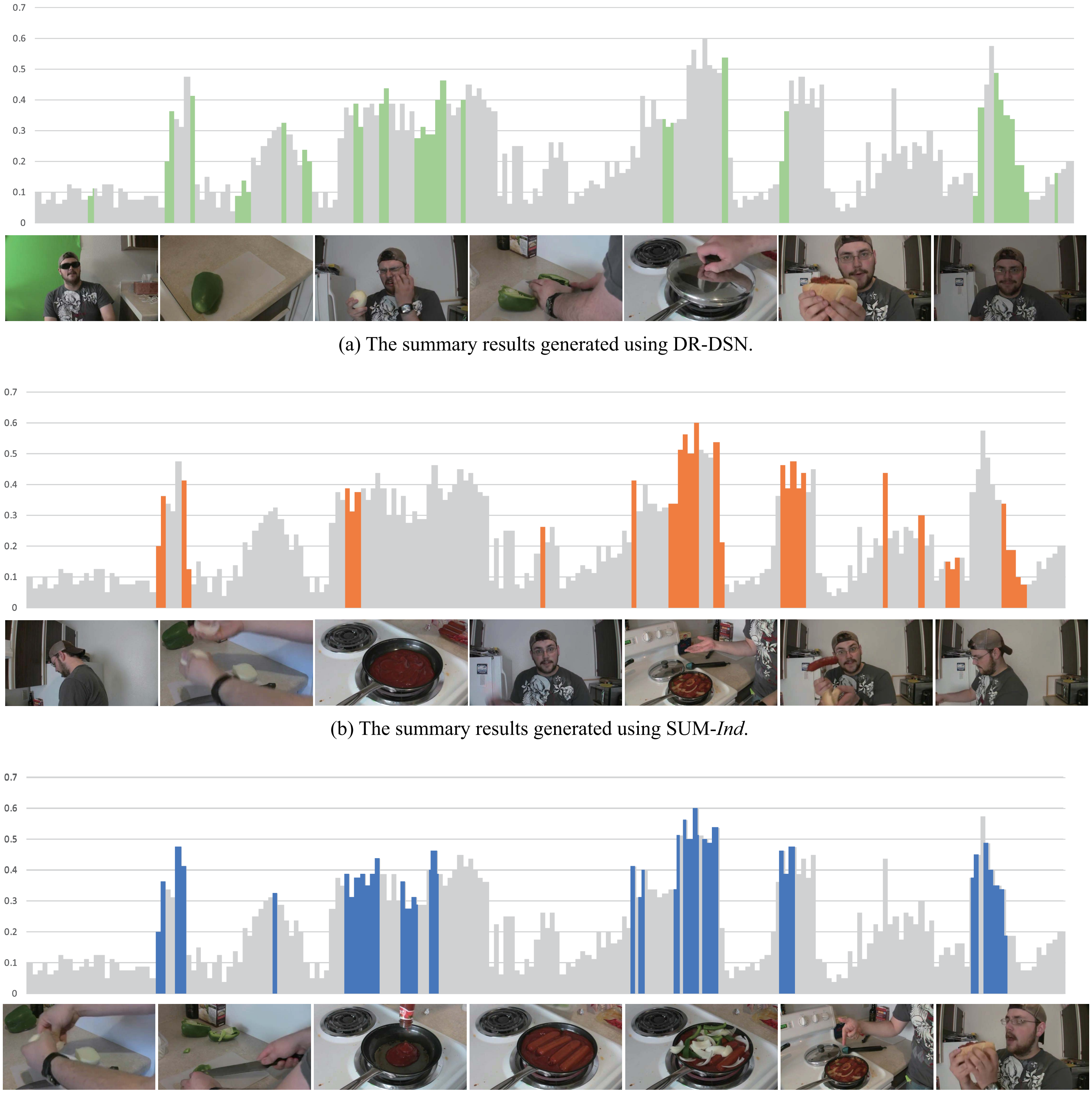
### D. Qualitative Evaluation

We performed a qualitative analysis of the performance of the different methods on the same video. As shown in Figure 6, we compare the state-of-the-art SUM-Ind [59] method and the DR-DSN [26] method, which also applies the reward function. We used the 18th video in the TVSum dataset, named ’Poor Man’s Meals: Spicy Sausage Sandwich’. The video is 405 seconds long and has 9731 frames. It is a demonstration video of a man teaching how to make a spicy sausage sandwich. Various methods used this video as a sample for qualitative analysis because it contains multiple camera angles and the storyline is easy to understand. Therefore, we visualize the extracted results, where the user-annotated ground truth importance scores are in gray and the colored highlights are the outcomes for each algorithm.

The results of the DR-DSN [26] algorithm in Figure 6(a) show that this approach selects many portrait keyframes as video summaries. The selected shots are not able to string together a story that can be understood by the viewer. One of the most serious problems is that the algorithm uses the filmmaker as the focal point, so the summary contains many shots of people. Judging diversity rewards by comparing the similarity of two frames may be prone to confusion. For example, while the man is sitting in a different direction, DR-DSN will regard them as different keyframes. For the viewers, they perceive it as a similar clip.

Figure 6(b) shows the summary extraction of this video using the SUM-Ind [59] method. There are too many similar shots in the extracted results, such as the author sitting in front of the camera describing the sandwich making process. These shots are not enough to provide richer information for the viewer. Although this approach selects as many relevant frames as possible to enhance diversity, the redundant frames cause the storyline to be incomplete at the same time. Specifically, almost a few shots were attended to in the early stages of the video, which caused the loss of some crucial sandwich-making steps in the summary results.

Figure 6(c) shows the results of our proposed approach. It can be seen that our approach is more focused on shot-level semantics. There is no sensitivity to medium shots, rather more interest in close-up shots. Consequently, the process of chopping and cooking is preserved as much as possible. More importantly, although the shot semantics are filming techniques but match with human recognition. The summaries generated in this way not only solve the problem of information redundancy but also retain as much helpful information as possible. The extracted keyframes are distributed with all phases of the video and correspond to ground truth data with high



(a) The summary results generated using DR-DSN.

(b) The summary results generated using SUM-*Ind*.

(c) The summary results generated using our algorithm.

Fig. 6. The video summary of video #18 in the TVSum dataset. (a) is the visualization result using the DR-DSN algorithm, (b) is the result of using the SUM-Ind algorithm, and (c) is the result of using our approach.

importance scores. Obviously, the results that we extracted are more in line with the name of the film ’Poor Man’s Meals: Spicy Sausage Sandwich’. This demonstrates that our method can effectively summarize videos and yield satisfactory results.

## V. Conclusion

We propose an unsupervised deep learning approach based on shot-level semantic rewards for video summarization. We are inspired by the shooting techniques that need to be followed for video shooting and cleverly blend these shot semantics with unsupervised deep learning methods. The convolutional neural network serves as an encoder to extract the convolutional feature matrix in the video through the shot scene training dataset that we created. A bidirectional LSTM is used as a decoder to generate probability weights for keyframe selection, which helps to solve the problem of temporal dependency. In particular, we design a shot-level semantic reward function to generate representative and diverse video summaries. This reward function does not cause bias in summary extraction due to user subjectivity. The experiments on the classical datasets are sufficient to demonstrate the high accuracy and recall of our approach.

In the future, our work includes improvements and extensions to the model, such as using attention mechanisms. Certainly, adding information such as audio and subtitles to the reward function is a direction to be explored in the future.

## Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful suggestions and comments.

## References

[1] Z. Ji, K. Xiong, Y. Pang, and X. Li, "Video summarization with attention-based encoder–decoder networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1709–1717, Jun. 2019.

[2] Y. Yuan, T. Mei, P. Cui, and W. Zhu, "Video summarization by learning deep side semantic embedding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 226–237, Nov. 2017.

[3] J. Lei, Q. Luan, X. Song, X. Liu, D. Tao, and M. Song, "Action parsing-driven video summarization based on reinforcement learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 2126–2137, Jul. 2019.

[4] M. Ma, S. Mei, S. Wan, Z. Wang, D. D. Feng, and M. Bennamoun, "Similarity based block sparse subset selection for video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3967–3980, Oct. 2021.

[5] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "AC-SUM-GAN: Connecting actor-critic and generative adversarial networks for unsupervised video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3278–3292, Aug. 2021.

[6] C. Huang and H. Wang, "A novel key-frames selection framework for comprehensive video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 577–589, Feb. 2020.

[7] A. G. Money and H. Agius, "Video summarisation: A conceptual framework and survey of the state of the art," *J. Vis. Commun. Image Represent.*, vol. 19, no. 2, pp. 121–143, Feb. 2008.

[8] X. Xu, T. M. Hospedales, and S. Gong, "Discovery of shared semantic spaces for multiscene video query and summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 6, pp. 1353–1367, Jun. 2016.

[9] A. A. Khan, J. Shao, W. Ali, and S. Tumrani, "Content-aware summarization of broadcast sports videos: An audio–visual feature extraction approach," *Neural Process. Lett.*, vol. 52, no. 3, pp. 1945–1968, 2020.

[10] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan, "Large-scale video summarization using web-image priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2698–2705.

[11] Y. Cao *et al.*, "Recognize human activities from partially observed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2658–2665.

[12] M. H. R. Pereira, F. L. C. Pádua, D. H. Dalip, F. Benevenuto, A. C. M. Pereira, and A. M. Lacerda, "Multimodal approach for tension levels estimation in news videos," *Multimedia Tools Appl.*, vol. 78, no. 16, pp. 23783–23808, Aug. 2019.

[13] H. Joho, J. M. Jose, R. Valenti, and N. Sebe, "Exploiting facial expressions for affective video summarisation," in *Proc. ACM Int. Conf. Image Video Retr. (CIVR)*, Jul. 2009, pp. 1–8.

[14] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "TVSum: Summarizing web videos using titles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5179–5187.

[15] D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz, "Schematic storyboarding for video visualization and editing," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 862–871, Jul. 2006.

[16] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1346–1353.

[17] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a long video short: Dynamic video synopsis," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2006, pp. 435–441.

[18] S. Lee, J. Sung, Y. Yu, and G. Kim, "A memory network approach for story-based temporal summarization of 360° videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1410–1419.

[19] J. Shen and N. Robertson, "BBAS: Towards large scale effective ensemble adversarial attacks against deep neural network learning," *Inf. Sci.*, vol. 569, pp. 469–478, Aug. 2021.

[20] R. Panda, A. Das, Z. Wu, J. Ernst, and A. K. Roy-Chowdhury, "Weakly supervised summarization of web videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3657–3666.

[21] B. Zhao, X. Li, and X. Lu, "Hierarchical recurrent neural network for video summarization," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 863–871.

[22] B. Zhao, X. Li, and X. Lu, "HSA-RNN: Hierarchical structure-adaptive RNN for video summarization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7405–7414.

[23] L. Lebron Casas and E. Koblents, "Video summarization with LSTM and deep attention models," in *Proc. Int. Conf. Multimedia Modeling*. Thessaloniki, Greece: Springer, Dec. 2019, pp. 67–79.

[24] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, and P. Remagnino, "Summarizing videos with attention," in *Proc. Asian Conf. Comput. Vis.* Perth, WA, Australia: Springer, Jun. 2018, pp. 39–54.

[25] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial LSTM networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 202–211.

[26] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.

[27] E. Apostolidis, A. I. Metsai, E. Adamantidou, V. Mezaris, and I. Patras, "A stepwise, label-based approach for improving the adversarial training in unsupervised video summarization," in *Proc. 1st Int. Workshop AI Smart TV Content Prod., Access Del. (AI4TV)*, 2019, pp. 17–25.

[28] L. Yuan, F. E. Tay, P. Li, L. Zhou, and J. Feng, "Cycle-SUM: Cycle-consistent adversarial LSTM networks for unsupervised video summarization," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 9143–9150.

[29] M. Rochan and Y. Wang, "Video summarization by learning from unpaired data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7902–7911.

[30] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Unsupervised video summarization via attention-driven adversarial learning," in *Proc. Int. Conf. Multimedia Modeling*. Daejeon, South Korea: Springer, 2020, pp. 492–504.

[31] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *Proc. Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, 2014, pp. 540–555.

[32] M. Sun, A. Farhadi, and S. Seitz, "Ranking domain-specific highlights by analyzing edited videos," in *Proc. Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, 2014, pp. 787–802.

[33] G. Kim, L. Sigal, and E. P. Xing, "Joint summarization of large-scale collections of web images and videos for storyline reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4225–4232.

[34] V. Tiwari and C. Bhatnagar, "A survey of recent work on video summarization: Approaches and techniques," *Multimedia Tools Appl.*, vol. 80, no. 18, pp. 27187–27221, Jul. 2021.

[35] J. Almeida, R. D. S. Torres, and N. J. Leite, "Rapid video summarization on compressed video," in *Proc. IEEE Int. Symp. Multimedia*, Dec. 2010, pp. 113–120.

[36] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognit.*, vol. 30, no. 4, pp. 643–658, 1997.

[37] C. Li, Y.-T. Wu, S.-S. Yu, and T. Chen, "Motion-focusing key frame extraction and video summarization for lane surveillance system," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 4329–4332.

[38] E. Bulut and T. Capin, "Key frame extraction from motion capture data by curve saliency," in *Proc. 20th Annu. Conf. Comput. Animation Social Agents*, vol. 20, no. 5, Seoul, South Korea, 2007.

[39] M. Ajmal, M. Naseer, F. Ahmad, and A. Saleem, "Human motion trajectory analysis based video summarization," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 550–555.

[40] U. Damnjanovic, V. Fernandez, E. Izquierdo, and J. M. Martinez, "Event detection and clustering for surveillance video summarization," in *Proc. 9th Int. Workshop Image Anal. Multimedia Interact. Services*, 2008, pp. 63–66.

[41] A. Javed, A. Irtaza, H. Malik, M. T. Mahmood, and S. Adnan, "Multimodal framework based on audio-visual features for summarisation of cricket videos," *IET Image Process.*, vol. 13, no. 4, pp. 615–622, 2019.

[42] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 766–782.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[44] A. Kulesza and B. Taskar, "Determinantal point processes for machine learning," 2012, *arXiv:1207.6083*.

[45] Y.-T. Liu, Y.-J. Li, F.-E. Yang, S.-F. Chen, and Y.-C.-F. Wang, "Learning hierarchical self-attention for video summarization," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 3377–3381.

[46] Z. Ji, F. Jiao, Y. Pang, and L. Shao, "Deep attentive and semantic preserving video summarization," *Neurocomputing*, vol. 405, pp. 200–207, Sep. 2020.

[47] M. Rochan, L. Ye, and Y. Wang, "Video summarization using fully convolutional sequence networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 347–363.

[48] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[49] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.

[50] S. Lal, S. Duggal, and I. Sreedevi, "Online video summarization: Predicting future to better summarize present," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 471–480.

[51] Y. Yuan, H. Li, and Q. Wang, "Spatiotemporal modeling for video summarization using convolutional recurrent neural network," *IEEE Access*, vol. 7, pp. 64676–64685, 2019.

[52] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[53] M. Paul and M. M. Salehin, "Spatial and motion saliency prediction method using eye tracker data for video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1856–1867, Jun. 2019.

[54] W.-T. Chu and Y.-H. Liu, "Spatiotemporal modeling and label distribution learning for video summarization," in *Proc. IEEE 21st Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2019, pp. 1–6.

[55] T. Sebastian and J. J. Puthiyidam, "A survey on video summarization techniques," *Int. J. Comput. Appl.*, vol. 132, no. 13, pp. 30–32, Dec. 2015.

[56] Y. Shi, Z. Wei, H. Ling, Z. Wang, J. Shen, and P. Li, "Person retrieval in surveillance videos via deep attribute mining and reasoning," *IEEE Trans. Multimedia*, vol. 23, pp. 4376–4387, 2021.

[57] P. Tokmakov, M. Hebert, and C. Schmid, "Unsupervised learning of video representations via dense trajectory clustering," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 404–421.

[58] X. He *et al.*, "Unsupervised video summarization with attentive conditional generative adversarial networks," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 2296–2304.

[59] G. Yaliniz and N. Ikizler-Cinbis, "Using independently recurrent networks for reinforcement learning based unsupervised video summarization," *Multimedia Tools Appl.*, vol. 80, no. 12, pp. 17827–17847, May 2021.

[60] N. Gonuguntla *et al.*, "Enhanced deep video summarization network," in *Proc. BMVC*, 2019, pp. 1–9.

[61] S. Cai, W. Zuo, L. S. Davis, and L. Zhang, "Weakly-supervised video summarization using variational encoder–decoder and web prior," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 184–200.

[62] H.-I. Ho, W.-C. Chiu, and Y.-C. F. Wang, "Summarizing first-person videos from third persons' points of view," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 70–85.

[63] Y. Chen, L. Tao, X. Wang, and T. Yamasaki, "Weakly supervised video summarization by hierarchical reinforcement learning," in *Proc. ACM Multimedia Asia*, Dec. 2019, pp. 1–6.

[64] M. Gygli, H. Grabner, H. Riemenschneider, and L. V. Gool, "Creating summaries from user videos," in *Proc. Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, 2014, pp. 505–520.

[65] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh, "Gaze-enabled egocentric video summarization via constrained submodular maximization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2235–2244.

[66] C. Bi *et al.*, "Dynamic mode decomposition based video shot detection," *IEEE Access*, vol. 6, pp. 21397–21407, 2018.

[67] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[68] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[69] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, May 1992.

[70] W.-S. Chu, Y. Song, and A. Jaimes, "Video co-summarization: Video summarization by visual co-occurrence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3584–3592.

[71] K.-H. Zeng, T.-H. Chen, J. C. Niebles, and M. Sun, "Generation for user generated videos," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 609–625.

[72] Y. Zhang, X. Liang, D. Zhang, M. Tan, and E. P. Xing, "Unsupervised object-level video summarization with online motion auto-encoder," *Pattern Recognit. Lett.*, vol. 130, pp. 376–385, Feb. 2020.

[73] Y. Jung, D. Cho, D. Kim, S. Woo, and I. S. Kweon, "Discriminative feature learning for unsupervised video summarization," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 8537–8544.

[74] P. Li, Q. Ye, L. Zhang, L. Yuan, X. Xu, and L. Shao, "Exploring global diverse attention via pairwise temporal relation for video summarization," *Pattern Recognit.*, vol. 111, Mar. 2021, Art. no. 107677.

[75] B. Zhao, X. Li, and X. Lu, "Property-constrained dual learning for video summarization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 3989–4000, Oct. 2020.

[76] S. E. F. De Avila, A. P. B. Lopes, A. Da Luz, Jr., and A. De Albuquerque Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognit. Lett.*, vol. 32, no. 1, pp. 56–68, Jan. 2011.

[77] B. Zhao and E. P. Xing, "Quasi real-time summarization for consumer videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2513–2520.

[78] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1059–1067.

**Ye Yuan** received the bachelor's and master's degrees from the School of Computer Software, Tianjin University, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the College of Intelligence and Computing. His research interests include multimedia applications, graphic processing, and visual analysis.

**Jiawan Zhang** (Senior Member, IEEE) received the master's and Ph.D. degrees in computer science from Tianjin University, in 2001 and 2004, respectively. He is currently a Professor with the School of Computer Software and an Adjunct Professor with the School of Computer Science and Technology, Tianjin University. He holds five patents, five software copyrights, and published more than 50 academic papers in peer-reviewed journals and conferences. His main research interests include computer graphics and realistic image synthesis. He was a recipient of the IBM Global Faculty Award twice, for his contribution to the software industry, in 2005 and 2006. He was the Conference Chair of the ACM VINCI Conference in 2013 and the Co-Chair of IEEE PacificVis VAST Workshop from 2015 to 2016.