INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research

International Journal of Computer Applications in Technology

ISSN online: 1741-5047 - ISSN print: 0952-8091
https://www.inderscience.com/ijcat

An unsupervised video summarisation method based on temporal convolutional networks

Ke Jin, Hui Li, Haoran Li, Qichuang Liu, Rong Chen, Shikai Guo

DOI: 10.1504/IJCAT.2024.10067536

Article History:

Received:	11 September 2023
Last revised:	13 November 2023
Accepted:	18 January 2024
Published online:	27 August 2025

Copyright © 2025 Inderscience Enterprises Ltd.

An unsupervised video summarisation method based on temporal convolutional networks

Ke Jin, Hui Li*, Haoran Li, Qichuang Liu, Rong Chen and Shikai Guo

School of Information Science and Technology,
Dalian Maritime University,
Dalian, Liaoning, China
Email: jin_ke@dlmu.edu.cn
Email: toy@dlmu.edu.cn
Email: li_hui@dlmu.edu.cn
Email: liuqichuang@dlmu.edu.cn
Email: rchen@dlmu.edu.cn
Email: shikai.guo@dlmu.edu.cn
*Corresponding author

Abstract: Video summarisation automatically selects a sparse subset of video frames that best represent the semantic content of the input video. Previous work mainly utilised Long Short-Term Memory (LSTM) networks to learn how to assess the importance of each video frame and then select appropriate frames to compose a video summary. However, these models still have shortcomings, such as limited memory capacity in handling long-term dependencies, and extended training time. Therefore, we present a deep video summarisation model, named TCN-SUM, which centres around Bi-TCN and models the frame sequence. TCN-SUM consists of three modules: frame selection module incorporates both Bidirectional Temporal Convolutional Network (Bi-TCN) and self-attention to model inter-frame dependencies, video reconstruction module reconstructs the original video based on the summary, and discriminator module measures the similarity between the original and reconstructed videos. Experimental studies on two benchmark data sets demonstrate that TCN-SUM outperforms state-of-the-art techniques, achieving superior performance in unsupervised approaches and showing competitiveness compared to supervised methods.

Keywords: Bi-TCN; video summarisation; self-attention; LSTM.

Reference to this paper should be made as follows: Jin, K., Li, H., Li, H., Liu, Q., Chen, R. and Guo, S. (2025) 'An unsupervised video summarisation method based on temporal convolutional networks', *Int. J. Computer Applications in Technology*, Vol. 76, Nos. 1/2, pp.1–16.

Biographical notes: Ke Jin received the master's degree in computer technology from Dalian Maritime University, Dalian, China in 2024. He is a graduate of the School of Information Science and Technology of Dalian Maritime University. His current research interests include video summarisation, visual algorithm testing, and software test development.

Hui Li received the PhD degree in computer architecture from Northeastern University, Shenyang, China in 2013. He is an Associate Professor with the School of Information Science and Technology, Dalian Maritime University. His current research interests include deep learning, image processing, and intelligent software engineering.

Haoran Li received a bachelor's degree in resource exploration engineering from the Earth Sciences and Engineering College, Shandong University of Science and Technology, Shandong, China, in 2018. He is currently pursuing the MS degree in software engineering major from the Information Science and Technology College, Dalian Maritime University, Dalian, China. His research interests include computer vision, video object detection.

Qichuang Liu received the Bachelor's degree in Computer Science and Technology from Hubei University in Wuhan, China, in 2022. He is currently a postgraduate student at the School of Information Science and Technology, Dalian Maritime University, with a research focus on video deblurring.

Rong Chen (Member, IEEE) received the MS and PhD degrees in computer software and theory from Jilin University, Changchun, China, in 1997 and 2000, respectively. He is currently a Professor with the College of Information Science and Technology, Dalian Maritime University,

Copyright © 2025 Inderscience Enterprises Ltd.

Dalian, China. He was previously with Sun Yat-sen University, Guangzhou, China. His research interests include software diagnosis, collective intelligence, activity recognition, and Internet and mobile computing.

Shikai Guo received the BS degree in computer science and technology from Liaoning University, Shenyang, China, in 2011, and the PhD degree in computer applications technology from Dalian Maritime University, Dalian, China, in 2018. He is currently an Associate Professor with the College of Information Science and Technology, Dalian Maritime University, Dalian, China. His current research interests include intelligent software engineering, software testing, and program analysis techniques.

1 Introduction

On July 2015, YouTube revealed that it receives over 400 hours of video content every minute, equivalent to 65.7 years of continuous watching in a single day. Till now, YouTube's intake of video content per minute has surpassed 1000 hours, encompassing a wide range of content such as news videos, entertainment videos, trending videos and more. However, YouTube is just one of the video-sharing platforms; there are numerous domestic platforms in countries like China, including Bilibili, TikTok and Kuaishou. So, how can people efficiently navigate through this vast sea of videos and discover content that interests them? This is where automatic video summarisation technology comes into play. This technology enables the generation of concise summaries that convey the essential parts of the original video. As a result, it aids viewers in browsing and finding the videos they need, potentially enhancing audience engagement and content consumption to some extent.

The application fields of automatic video summarisation technology are very extensive (Zhang et al., 2016; Zhao et al., 2018; Ji et al., 2020), encompassing but not limited to generating highlights of a sports match (such as basketball dunks, soccer goals); creating trailers for movies or TV shows; extracting summary videos from surveillance footage (capturing significant events) to facilitate efficient monitoring. As a result, automatic video summarisation technology has become an indispensable tool that has emerged with the progression of time. The objective of video summarisation is to produce a concise yet comprehensive video summary to enable efficient browsing experiences. An ideal video summary aims to provide users with the maximum information from the original video in the shortest possible time. Moreover, it serves as a foundational step for many other practical applications, including video indexing, video retrieval and event detection.

Numerous previous studies on video summarisation have already been presented. Earlier works focused on various attributes of videos, such as target scenes, types of video content and granularity of video segments. These models often overlooked the inherent temporal dependencies between video frames, resulting in generated summaries that frequently fell short of achieving advanced levels of performance. In recent years, deep learning methods represent the cutting-edge of video summarisation (Sabha and Selwal, 2023; Meena et al., 2023; Hu et al., 2023; Sreeja and Kovoor, 2023). Particularly, with the emergence of LSTM, the technological landscape of automatic video summarisation has witnessed a significant leap. This is attributed to the fact that the video summarisation task can be viewed as a sequence modelling problem, and LSTM inherently excels at such tasks. These methods can be roughly divided into supervised, unsupervised and weakly supervised methods.

Supervised methods (Zhang et al., 2016; Zhao et al., 2018; Ji et al., 2020) rely on data sets containing manually generated ground-truth summaries. Based on these data sets, they aim to uncover the underlying patterns of video summarisation. However, generating ground-truth data is a time-consuming and laborious task. Additionally, the subjectivity of video summarisation can result in drastically different summaries for the same video, making it difficult to train methods that use these summaries as ground truth.

Unsupervised methods attempt to learn video summarisation without using real data. Some of these methods rely on Generative Adversarial Networks (GAN) to seek a solution to assess the representativeness of any generated summary (Meena et al., 2023; Hu et al., 2023). Others build upon reinforcement learning (Sreeja and Kovoor, 2023; Yuan and Zhang, 2023), and define rewards based on desired features of video summarisation, like the diversity of its visual content. Many of these methods employ LSTM units to learn how to assess the importance of each video frame.

Additionally, some researchers have proposed weakly supervised methods to address the problem of video summarisation. The purpose of proposing these methods is to reduce the reliance on ground-truth annotations. Compared to using manually labelled true annotations, these methods simply use low-cost labels such as subtitles and titles for learning.

Recently, Transformer-based video summarisation methods have emerged (Hsu et al., 2023; Chen et al., 2022; Zhao et al., 2022; Feng et al., 2022). While these methods achieve impressive summary results, their lengthy training times remain a significant challenge. Therefore, LSTM-based models are still the common used solutions for video summarisation.

However, LSTM-based video summarisation techniques still encounter certain issues:

1) LSTM lacks parallelisability, resulting in slow training speeds. LSTM, being a variant of RNN, can only process sequentially. Specifically, LSTM processes one video frame at a time, with the current frame's processing dependent on the previous frame's LSTM computation. Although bidirectional LSTMs exist, both directions follow the aforementioned sequential flow, offering no significant acceleration. Considering that videos often consist of hundreds or thousands of frames, the inherent characteristics of recurrent network architecture make efficient processing of these videos challenging, posing an issue yet to be resolved.
2) LSTM continues to face limitations in addressing long-term dependency problems due to inadequate memory capacity. In LSTM, each hidden state retains only partial information from its inputs and previous hidden states. Consequently, the central hidden state might retain very little information from distant states. This inability to perfectly capture long-term dependencies from a longer video may result in a decline in the quality of summaries.

To address these issues, we propose a video summarisation method based on TCN (Bai et al., 2018). Compared to LSTM-based sequential learning models, the TCN-based approach offers the following advantages:

1) TCN models long sequences through dilated convolutions, where the dilation factor grows exponentially with the layers $(d = 1, 2, 4, ...)$. This implies that by stacking several layers of dilated convolutions, the ability to model extremely long sequences can be achieved theoretically.
2) TCN incorporates residual connections and weight normalisation. The combination of these two techniques addresses common issues in RNNs. For instance, when modelling long sequences, TCN avoids the problem of vanishing or exploding gradients that affect RNNs, thereby mitigating the issue of fading long-term memory.
3) The convolutional structure of TCN leverages the benefits of GPU parallel processing, leading to faster training speeds and improved training results.
4) TCN can be easily extended to a bidirectional structure, referred to as Bi-TCN. In our subsequent work, we adopt the Bi-TCN structure as the main architecture for our frame selection module.

In summary, the contributions of this work are as follows:

- We utilise a variant of the Bi-TCN convolutional model to address video summarisation challenges;
- We introduce a self-attention mechanism embedded within the Bi-TCN to enhance the connections between video frames;
- We propose an architecture named TCN-SUM, which combines Bi-TCN with GAN to learn keyframe selection strategies in a fully unsupervised manner;
- We compare our approach with advanced methods, showcasing its outstanding competitiveness.

In summary, our work presents a novel approach to video summarisation, leveraging the Bi-TCN model, self-attention, and unsupervised learning through the TCN-SUM architecture, all of which contribute to achieving competitive results in this field.

2 Related work

With the advancement of deep learning, automatic video summarisation technology has experienced rapid progress. Among these developments, RNNs, especially LSTM, excel in modelling long-term dependencies, aligning perfectly with the concepts of video summarisation. Previous deep learning-based methods can be broadly categorised into three groups:

1) *Supervised methods*: These methods rely on labelled data with ground-truth summaries for training. They learn to generate summaries that align with human-annotated summaries.
2) *Unsupervised methods*: These approaches do not require ground-truth summaries for training. Instead, they aim to discover patterns within the video data to generate summaries without explicit human guidance.
3) *Weakly supervised methods*: These techniques fall between supervised and unsupervised methods. They leverage partial annotations, such as video titles or timestamps, to guide the summarisation process without relying on full summaries.

Each of these categories offers unique advantages and challenges in the quest to enhance the quality and efficiency of automatic video summarisation.

2.1 Supervised video summarisation

Zhang et al. (2016) introduced LSTM to video summarisation, in which video summarisation is taken as a structured prediction task on sequential data. They employed Bi-LSTM (Graves and Schmidhuber, 2005) to model the temporal dependencies between video frames. Additionally, they employed a Multi-Layer Perceptron (MLP) to estimate importance scores for these frames. Furthermore, the paper demonstrated how to generate visually diverse summaries by combining Bi-LSTM with Determinantal Point Processes (Kulesza and Taskar, 2012). Subsequently, Zhao et al. (2018) introduced a hierarchical and adaptive RNN structure comprising two LSTM layers. The first layer segments the video into multiple shots, while the second layer learns shot-level information and obtains corresponding scores. Ji et al. (2020) formulated the supervised video summarisation problem as a sequence-to-sequence learning task. In this setup, the input consists of an original sequence of video frames, and the output is a sequence of key shots. They introduced an LSTM-based encoder-decoder network named AVS, incorporating an attention mechanism. Additionally,

they developed two methods for generating video summaries: one based on additive attention mechanism, denoted as A-AVS, and another based on multiplicative attention mechanism, known as M-AVS. Fajtl et al. (2019) introduced an computationally efficient video summarisation method based on a soft self-attention mechanism for sequence-to-sequence transformation. The aim was to avoid the use of computationally demanding LSTM. This network is composed of a soft self-attention mechanism and a two-layer fully connected network designed for regressing importance scores of frames.

To overcome the slow training drawback of LSTM, Rochan et al. (2018) redefined video summarisation as a semantic segmentation task and introduced a Fully Convolutional Sequence Network. The network structure relies on convolutional and deconvolutional layers, with an increasing effective context size as the network deepens. This enables the network to effectively model long-term dependencies between frames and learn frame importance.

In supervised video summarisation, there is a challenge related to the scarcity of ground-truth annotations in data sets, coupled with the high cost of creating such annotations. To address this scarcity, Jiang and Mu (2022) introduced a collaborative teaching network named iPTNet, based on importance propagation. This approach establishes a connection between video summarisation and video moment localisation, enabling cross-task sample transfer. They designed an importance propagation module to convert the importance maps from the video summarisation module to those from moment localisation. This mechanism leverages data from another task to optimise one of the tasks, thereby mitigating the issue of limited annotations.

Simultaneously, some supervised methods utilise GAN (Goodfellow et al., 2014) to tackle video summarisation. The summariser (corresponding to the generator in GAN) takes a sequence of video frames as input and generates summaries by calculating frame-level importance scores. Fu et al. (2019) introduced a GAN-based training framework that combines the strengths of both unsupervised and supervised video summarisation methods. The generator is an attention-aware Ptr-Net, generating cut points (start and end points) for summary segments. The discriminator is a 3D-CNN classifier that distinguishes whether a segment comes from user-annotated summaries or generated summaries. The discriminator's output serves as a reward to train the generator using reinforcement learning, rather than the conventional adversarial loss. To date, GAN-based approaches in supervised video summarisation are limited, but GAN-based methods are widely applied in unsupervised video summarisation.

2.2 Unsupervised video summarisation

Supervised learning approaches for video summarisation utilise ground-truth labels representing importance scores for each frame to train deep neural networks. Because these models are trained on manually annotated data, they can explicitly learn semantic features. However, annotating many video frames is costly, and overfitting can occur when labelled data is scarce. These limitations can be mitigated by employing unsupervised learning methods.

Mahasseni et al. (2017) introduced the pioneering approach of using Variational Autoencoders (VAE) (Kingma and Welling, 2014) and GAN to learn video summarisation without manual annotations. They combined an LSTM-based keyframe selector with a VAE and a trainable discriminator. Through an adversarial learning process, their aim was to minimise the distance between the original video and its reconstructed version based on the summary. Building on (Mahasseni et al., 2017) network architecture, Apostolidis et al. (2020) proposed a progressive, label-based approach to train the adversarial component of the network, enhancing summarisation performance. Yuan et al. (2019) introduced a novel unsupervised video summarisation model utilising a novel Cycle-Consistent Adversarial LSTM architecture (Zhu et al., 2017). This architecture effectively maximises mutual information between summary videos and original videos. The framework consists of a frame selector and a cycle-consistent evaluator. The frame selector (Bi-LSTM) predicts importance scores for each frame, selecting frames with high importance scores to form the summary video. The cycle-consistent evaluator assesses the quality of the selected frames through cycle reconstruction and guides the selector to identify frames with maximal information content for the summary video. The evaluator comprises two GANS: a forward GAN learns to reconstruct the original video from the summary video, while a backward GAN attempts to learn reverse reconstruction from the original video to the summary video.

Jung et al. (2019) addressed and resolved two key issues present in previous unsupervised methods: (1) the ineffective feature learning caused by the flat distribution of importance scores predicted for each frame and (2) the training challenges when dealing with long video inputs. To tackle the first issue, they proposed a variance loss that allows the network to predict output scores with high variances for each frame, enabling effective feature learning and significantly improving model performance. For the second issue, they constructed a Block and Stride Network (CSNet) and integrated a diversity attention mechanism to efficiently handle videos of varying lengths and focus on dynamic scenes related to key shots. He et al. (2019) introduced a self-attention-based conditional GAN network (Mirza and Osindero, 2014). In this network, the summary generator generates weighted features for each frame's temporal representation and predicts frame-level importance scores. The discriminator attempts to distinguish between the weighted frame features and original frame features. Additionally, a conditional feature selector is employed to guide the GAN model's attention to more important temporal regions within the entire video frame. Notably, this work was the first to introduce frame-level multi-head self-attention (Vaswani et al., 2017) for video summarisation. This mechanism learns the long-term temporal dependencies of the entire video sequence, overcoming the local constraints of recurrent units like LSTM.

Rochan and Wang (2019) introduced a novel approach for learning video summarisation from unpaired data. They used a FCSN as a summariser to generate summaries and employed the encoder part of FCSN as a discriminator to differentiate between generated summaries and human-created summaries. The goal of this method was to learn a mapping function from original videos to human-created summaries, so that the distribution of generated summary videos becomes similar to that of human-created summaries with the help of adversarial objectives. Additionally, diversity constraints were applied to the generated summaries to ensure visual diversity. Jung et al. (2020) proposed an input decomposition strategy aimed at capturing both local and global information from input video frames. They introduced a Self-Attention Embedding module and a Self-Attention module with Relative Position Embedding. This self-attention module effectively captured long-term dependencies among video frames. Moreover, by leveraging the input decomposition strategy, the computational complexity of this self-attention module was significantly reduced, improving computational efficiency and further enhancing summary performance.

Zhou et al. (2018) was the first work to apply reinforcement learning to unsupervised video summarisation. They framed video summarisation as a sequential decision process, developed a Deep Summary Network (DSN) to generate video summaries, and designed a novel diversity-representativeness reward function. The diversity reward measured the dissimilarity between selected frames, while the representativeness reward calculated the distance between frames and their nearest selected frames. The reward function assessed the diversity and representativeness of the generated summaries, while the DSN aimed to achieve higher rewards by learning to generate more diverse and representative summaries. Apostolidis et al. (2021) introduced an innovative architecture that embedded an Actor-Critic model into a GAN. This approach learned keyframe selection and summarisation strategies in a completely unsupervised manner and used feedback from the discriminator to train the Actor-Critic model. It eliminated the need for external supervision or handcrafted rewards, as it automatically learned keyframe selection strategies based on the feedback from a trainable discriminator.

Apostolidis et al. (2022) pointed out several limitations of unsupervised methods, including the unstable training of generator-discriminator architectures, the limited memory capacity of RNNs for modelling long-term dependencies in frames and the inability of RNN-based architectures to parallelise training. To address these limitations of unsupervised methods, they introduced an approach that relies solely on a self-attention mechanism to estimate the importance of video frames, known as the Concentrated Attention Mechanism (CAM). This method aimed to focus on non-overlapping blocks along the main diagonal of the attention matrix, using a self-attention mechanism. By extracting and leveraging unique characteristics and diversity-related information about video frames, the cam enriched the existing information. This led to better estimation of the importance of different parts of the video and significantly reduced the number of learnable parameters.

2.3 Weakly supervised video summarisation

In addition to supervised and unsupervised approaches, a subset of researchers have introduced weakly supervised methods to address the problem of video summarisation. The purpose behind proposing such methods is also to reduce the reliance on ground-truth annotations. Unlike the use of manually labelled ground-truth annotations, this approach relies on lower-cost labels such as subtitles and captions for learning. Panda et al. (2017) proposed the first method to introduce weak supervision into video summarisation. They introduced a flexible deep 3D-CNN architecture that defines video classification using only video metadata, without using any manually curated training data. Specifically, the main idea of this method is to utilise multiple videos from a category to automatically learn parameterised models for video classification. These models are then used to classify given videos and identify the most significant segments influencing the model's output. Cai et al. (2018) introduced a VAE model with an attention mechanism for learning latent semantics from web videos. They also proposed a sequence encoder-decoder framework. The VAE works by mapping videos to a continuous latent space through an encoder, and then reconstructing input videos from samples of the latent variables using a decoder. They further introduced a weakly supervised semantic matching loss to learn title-relevant summaries. Ho et al. (2018) presented a novel deep neural network for summarising first-person videos. This approach leverages principles of transfer learning and utilises annotated third-person videos to learn how to summarise first-person videos.

3 Proposed method

Video summarisation aims to create a sparse and compact version of the original video by selecting the most representative and crucial segments of the source material. The generated summary typically consists of a set of representative video frames or clips, which are concatenated in chronological order to form a shorter video.

3.1 Problem formulation

In our research, we approach video summarisation as a sequence modeling problem. Specifically, given an original video $V = \{v_i\}_{i=1}^{T}$ (where T represents the number of frames in the video), we first utilise a feature extraction module to obtain features $X = \{x_i\}_{i=1}^{T}$ for each frame. These frame features are then passed through a linear compression layer to reduce their dimensionality to 500, resulting in representation

$X' = \{x_i'\}_{i=1}^{T}$. Subsequently, these compressed features are input into a frame selection module, which calculates importance scores $S = \{s_i\}_{i=1}^{T}, s_i \in [0,1]$ for each frame. Additionally, residual connections are employed to maintain the temporal order of frames in the original video. Following this, frame-level importance scores are used to weight each frame's feature, resulting in weighted features $W = \{w_i\}_{i=1}^{T} = \{x_j' \cdot s_i\}_{i=1}^{T}$. Afterward, the weighted features are fed into a video reconstructor, where an inverse process is employed to regenerate a reconstruction of the original video. This reconstructed video is then compared with the original video using a discriminator for classification. This iterative process continues until the generator is capable of producing reconstructed videos that confuse the discriminator, rendering the discriminator unable to accurately distinguish between the reconstructed and original videos. This adversarial training loop is designed to enhance the generator's ability to generate realistic video summaries.

3.2 Overall network model

To address the challenge of video summarisation, we introduce a video summarisation architecture based on TCN, referred to as TCN-SUM, as depicted in Figure 1. The model comprises three key components:

- *Data preprocessing module*: This stage prepares appropriate frame feature sequences from the raw video input.
- *Frame selection module*: In this phase, importance scores for each frame are computed. These scores reflect the significance of each frame within the entire video. A summary is then generated based on these importance scores.
- *Video reconstruction module*: This module attempts to reconstruct a video that confounds the discriminator based on the generated summary.
- *Discriminator module*: The discriminator endeavours to differentiate between the original video and the reconstructed video generated from the summary.

The overall architecture of TCN-SUM is described as follows: First, the original video is processed through a data pre-processing module to obtain a feature sequence shaped as $T \times d$, where T represents the number of video frames, and d represents the dimension of the feature vector for each video frame; second, this feature sequence is input into the frame selection module to calculate the importance score for each frame; third, the feature vector for each frame is weighted by this importance score and fed into the generator module for the reconstruction of the original video finally, both the original and reconstructed videos are input into the discriminator module for evaluation. The generator and discriminator modules form a GAN architecture, engaging in adversarial training.

3.3 Data pre-processing module

The purpose of the data pre-processing module is to extract suitable frame feature sequences from the original video. To ensure a fair comparison with other methods, we adopt the same feature extraction technique as they chose. First, the original video is downsampled to 2FPS. The downsampled video frames are then input into the GoogLeNet model, which has been pre-trained on ImageNet. The output from the pool5 layer of GoogLeNet is used as the visual feature for each video frame, thereby obtaining a 1024-dimensional feature vector for each frame. Following the feature extraction method provided by Zhang et al. (2016), we generated an h5 file for each data set. In order to understand our method and the problems to be solved more clearly, we give the structure of the h5 file in Table 1, which mainly gives the various parameters of the extracted video feature sequence and the meaning they express. After the video features are extracted, we also use a linear compression layer to reduce the feature vector from 1024 dimensions to 500 dimensions to speed up the training process.

Figure 1 The overall architecture of TCN-SUM (see online version for colours)

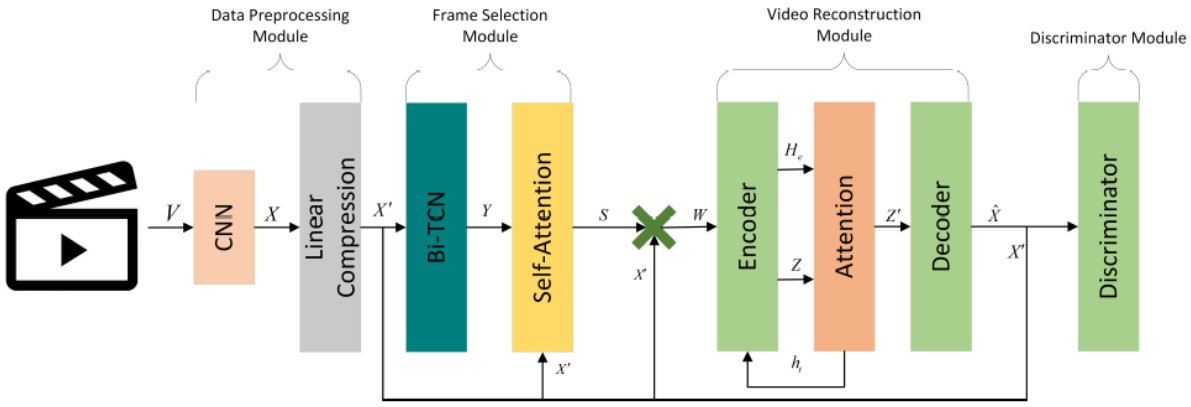


Table 1 Structure of each data set h5 file

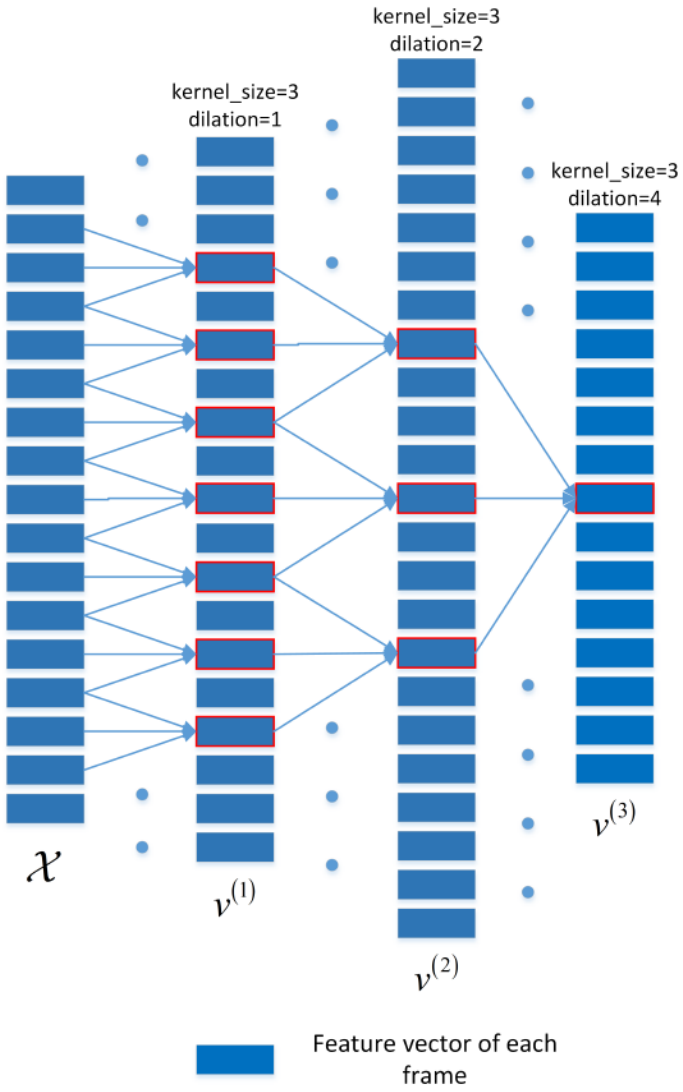
Parameter	*Meaning*
features	2D-array with shape (n_steps, feature_dimension), feature sequence of each video
gtscore	1D-array with shape (n_steps), stores ground truth importance score
user_summary	2D-array with shape (num_users, n_frames), each row is a binary vector
n_steps	the number of frames after downsampling the original video
gtsummary	1D-array with shape (n_steps), ground truth summary provided by user
video_name	original video name

3.4 Frame selection module

The main role of the Frame Selection module is to choose appropriate frames to compose a summary. The Frame Selection module consists primarily of the Bi-TCN and Self-Attention components. Bi-TCN is utilised to establish long-term dependencies among video frames, while Self-Attention focuses on the more important segments within the video. Additionally, residual connections are employed to combine the original feature sequence from the video with the output of the Self-Attention module. This combined output then passes through layer normalisation, fully connected layers, as well as Rectified Linear Unit with Randomised Leaky Rectifier activation functions and Sigmoid activation functions. Below, a detailed explanation of the Frame Selection module is provided.

Bi-TCN: TCN is an emerging technique for modelling time series data. It addresses the issues of vanishing or exploding gradients during training that are commonly associated with RNNs. TCN has demonstrated promising results in a wide range of time series modelling problems. Its major advantage lies in its ability to perform parallel computations, making optimal use of computational resources. In order to apply TCN to video summarisation, we have devised a bidirectional TCN, referred to as Bi-TCN. Figure 2 illustrates the architecture of a three-layer Bi-TCN.

Figure 2 Three-layer structure of Bi-TCN (see online version for colours)



Specifically, each element of vector $v^{(3)}$ undergoes three layers of one-dimensional dilated convolution processing. This allows the integration of contextual information from both directions, forming a feature representation with long-term dependencies. In general, we input the video feature sequence X' into the Bi-TCN with n layers, and its output can be expressed as

$$\begin{cases} v^{(1)} = \theta_1\left(X', \delta^1, d^1\right) \\ \vdots \\ v^{(i)} = \theta_i\left(v^{(i-1)}, \delta^i, d^i\right) \\ \vdots \\ v^{(n)} = \theta_n\left(v^{(n-1)}, \delta^n, d^n\right) \end{cases} \tag{1}$$

where θ_i represents the one-dimensional dilated convolution of the i-th layer, δ^i and d^i denote the convolutional kernel size and dilation factor, respectively. Finally, $Y = v^{(n)}$ corresponds to the long-term dependency representation of the desired local features.

Self-attention: The Self-Attention mechanism has demonstrated its effectiveness in handling time series problems. As a result, we have incorporated the Self-Attention mechanism into our model. Specifically, after obtaining refined features, we employ an attention-based network to capture crucial segments within the video, which are then utilised to predict the ultimate importance scores. The architecture of the attention network is illustrated in Figure 3.

The Self-Attention mechanism models dependencies between the desired output and each input. We feed the feature sequence output by the Bi-TCN into the Self-Attention mechanism, and compute Query, Key and Value components separately. Their computation processes are

$$\begin{cases} Q = YW^Q \in \mathbb{R}^{T\times d} \\ K = YW^K \in \mathbb{R}^{T\times d} \\ V = YW^V \in \mathbb{R}^{T\times d} \end{cases} \tag{2}$$

where d represents the dimension of the features, and $W^Q \in \mathbb{R}^{d\times d}$, $W^K \in \mathbb{R}^{d\times d}$ and $W^V \in \mathbb{R}^{d\times d}$ are weight matrices to be learned. Then, we compute the dot product of Query and Key values, resulting in an attention map. Subsequently, we apply a row-wise SoftMax function to obtain normalised weights reflecting inter-frame correlations. The result of the SoftMax function is then multiplied by the Value matrix V. Finally, a residual connection is employed to ensure that this feature sequence maintains the temporal sequence attributes of the original video. The specific process can be expressed as:

$$H = Atten(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{3}$$

$$Y' = X' + HW^Z \tag{4}$$

where $W^Z \in \mathbb{R}^{d\times d}$. Then, it undergoes layer normalisation, followed by fully connected layers, RReLU activation function and Sigmoid activation function, resulting in predicted frame-level importance scores, which can be represented as

$$S = Sigmoid\left(RRelu\left(FC\left(LayerNorm\left(Y'\right)\right)\right)\right) \tag{5}$$

Finally, the predicted frame-level importance scores S are used to weight the original features X', denoted as W. This weighted result is then fed into the video reconstruction module to reconstruct the original visual video.

3.5 Video reconstruction module

The purpose of the video reconstruction module is to attempt the reconstruction of videos that can confuse the discriminator. In this module, we adhere to the network architecture designed in Apostolidis et al. (2020), referring to this module as the Adversarial Autoencoder (AAE), as illustrated in Figure 4.

Figure 3 The structure of self-attention (see online version for colours)

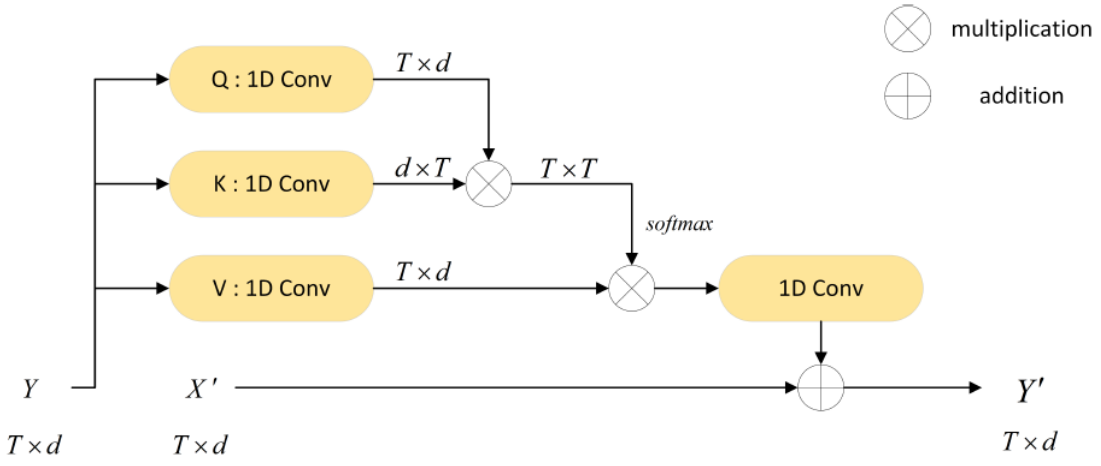
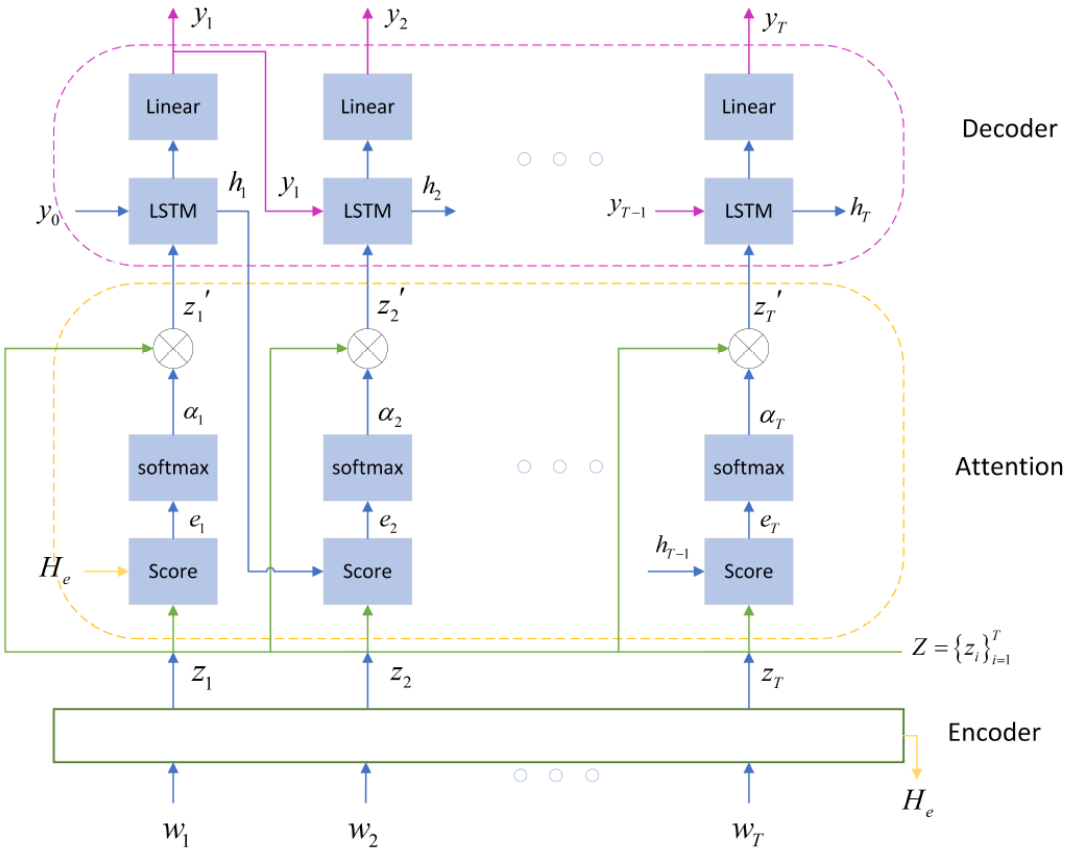


Figure 4 The structure of AAE (see online version for colours)



Firstly, the weighted feature vector $W = \{w_t\}_{t=1}^{T}$ is fed into the encoder Bi-LSTM, referred to as $eLSTM$, to re-encode the weighted feature vector, resulting in output $Z = \{z_t\}_{t=1}^{T}$. This output is then used as input for the attention component, which can be represented as

$$Z = eLSTM(W) \tag{6}$$

For any time step t within the range $[2, T]$, the attention component takes the encoder's output $Z = \{z_t\}_{t=1}^{T}$ and the previous hidden state of the decoder, denoted as h_{t-1} and employs a scoring function to compute the attention energy vector e_t. This vector represents the correlation between the t-th frame of the video and the entire set of video frames. Finally, a softmax function is applied to normalise the attention energy vector, yielding the attention weight vector α_t.

For $t = 1$, as shown in Figure 4, the attention component uses the last hidden state H_e from the encoder as input since there is no previous hidden state from the decoder. Then, the attention weight vector α_t is multiplied by the encoder's output $Z = \{z_t\}_{t=1}^{T}$, resulting in a context vector $Z' = \{z'_t\}_{t=1}^{T}$. This can be represented as

$$z_t' = Z \otimes \alpha_t, t \in [1, T] \tag{7}$$

Subsequently, this context vector Z' is fed into the decoder, referred to as $dLSTM$, which combines it with the output y_{t-1} from the previous frame. This iterative process contributes to the step-by-step reconstruction of the video, expressed as

$$y_t = Linear\left(dLSTM\left(y_{t-1}, z_t'\right)\right) \tag{8}$$

Finally, we denote the reconstructed video as $\hat{X}$.

The scoring function we employ is a multiplication function, represented by

$$e_t^i = z_i^* W_a h_{t-1} \tag{9}$$

where z_i^* is the transposed encoder output of the i-th video frame, h_{t-1} is the decoder's hidden state at time step $t-1$, W_a represents learnable parameters, e_t^i stands for the unnormalised relevance score between the i-th video frame and the current frame, which is a scalar value denoting the correlation. The final attention weight a_t^i is computed through normalisation as

$$\alpha_t^i = \exp\left(e_t^i\right) / \sum_{j=1}^{T} \exp\left(e_t^j\right) \tag{10}$$

3.6 Discriminator module

The discriminator primarily consists of a Bi-LSTM aimed at distinguishing between the original feature X' and the reconstructed video's features $\hat{X}$. Specifically, our discriminator is composed of a Bi-LSTM with 500 hidden

units, referred to as *cLSTM* , followed by a fully connected layer and a sigmoid activation function. The input to the FC layer is the last hidden state of the LSTM. The sigmoid activation function generates a binary classification output. In practice, the label for the original feature X should be 1, and the label for the weighted feature X' should be 0. Once convergence is achieved, when the weighted feature becomes highly similar to the original feature, the discriminator struggles to accurately differentiate between the reconstructed and the original videos. At this point, the generator's output tends to resemble real data, indicating that the generated summary retains the essential information from the original video. The binary classification output is computed using the following equation:

$$prob = \text{sigmoid}\left(FC\left(cLSTM\left(X' or \hat{X}\right)\right)\right) \tag{11}$$

The variable $prob$ represents the likelihood that a video is either a reconstructed video or an original visual video. It's worth noting that the discriminator is employed solely during training. In testing, we forego both the discriminator and video reconstructor, relying solely on the frame selector. This frame selector takes the appearance features X' extracted from the video as input and assigns importance scores to each frame of the original video, facilitating further evaluation.

3.7 Loss function

Firstly, we introduce the reconstruction loss, aiming to minimise the information difference between the feature sequence $\hat{X}$ generated based on the summary and the original feature sequence X'. The intuition behind applying this loss is that a good summary should preserve the essential portions of the original video. Therefore, we calculate the reconstruction loss between the output $\phi(X')$ and $\phi\left(\hat{X}\right)$ from the last hidden layer of the discriminator LSTM, and the expression is

$$L_{recon}\left(X', \hat{X}\right) = \left\|\phi(X') - \phi\left(\hat{X}\right)\right\|_2 \tag{12}$$

where $\|\cdot\|_2$ represents the Euclidean distance.

Additionally, we also introduce a sparsity loss. The rationale behind incorporating this loss is that a good summary should be concise. The specific expression for this loss is

$$L_{sparsity}(S) = \left|\frac{1}{T}\sum_{t=1}^{T} s_t - \alpha\right| \tag{13}$$

where T is the total number of video frames, and α is a hyperparameter that controls the length of the summary. It penalises the selection of a large number of key frames in the summary.

Simultaneously, to regularise the training of the GAN, that is, ensuring that the discriminator classifies the reconstructed feature sequence $\hat{X}$ as summary and the original feature sequence X as the original video, we enforce the discriminator to learn to classify randomly generated summaries $\hat{X}_p$ as summaries. Here, $\hat{X}_p$ is a subset of video frames randomly selected by sampling from a given prior distribution and then reconstructed. Consequently, we define the prior loss as

$$L_{prior} = D_{KL}\left(q(e|x)\|p(e)\right) \tag{14}$$

where e is the hidden representation of x, $p(e)$ is used to map the values of the latent variable e to values that fit a standard normal distribution with mean 0 and variance 1, x is the observed data, $q(e|x)$ represents the probability of observing e given x, and D_{KL} denotes the Kullback-Leibler divergence.

Our objective is to find a better equilibrium between the generator and discriminator, which implies achieving improved video reconstruction by combining weighted frames and the learned data distribution through the variational autoencoder architecture. Consequently, we did not utilise the L_{GAN} loss from the model in Mahasseni et al. (2017), but instead followed the approach in Apostolidis et al. (2020) that employs a label-based method. In this method, the label '1' is assigned to the original video, while the label '0' is assigned to the video summary. To train the discriminator, we introduce the following two losses as:

$$L_{ORIG} = \left(1 - cLSTM\left(X'\right)\right)^2 \tag{15}$$

$$L_{SUM} = \left(cLSTM\left(\hat{X}\right)\right)^2 \tag{16}$$

where L_{ORIG} is used to minimise the mean squared error between the label of the original video and the probability computed when the discriminator is given the original video as input. Similarly, when the discriminator processes video reconstructions based on the summary, L_{SUM} is employed to minimise the MSE between the summary label and the calculated probability.

To train the generator, we introduce the following losses:

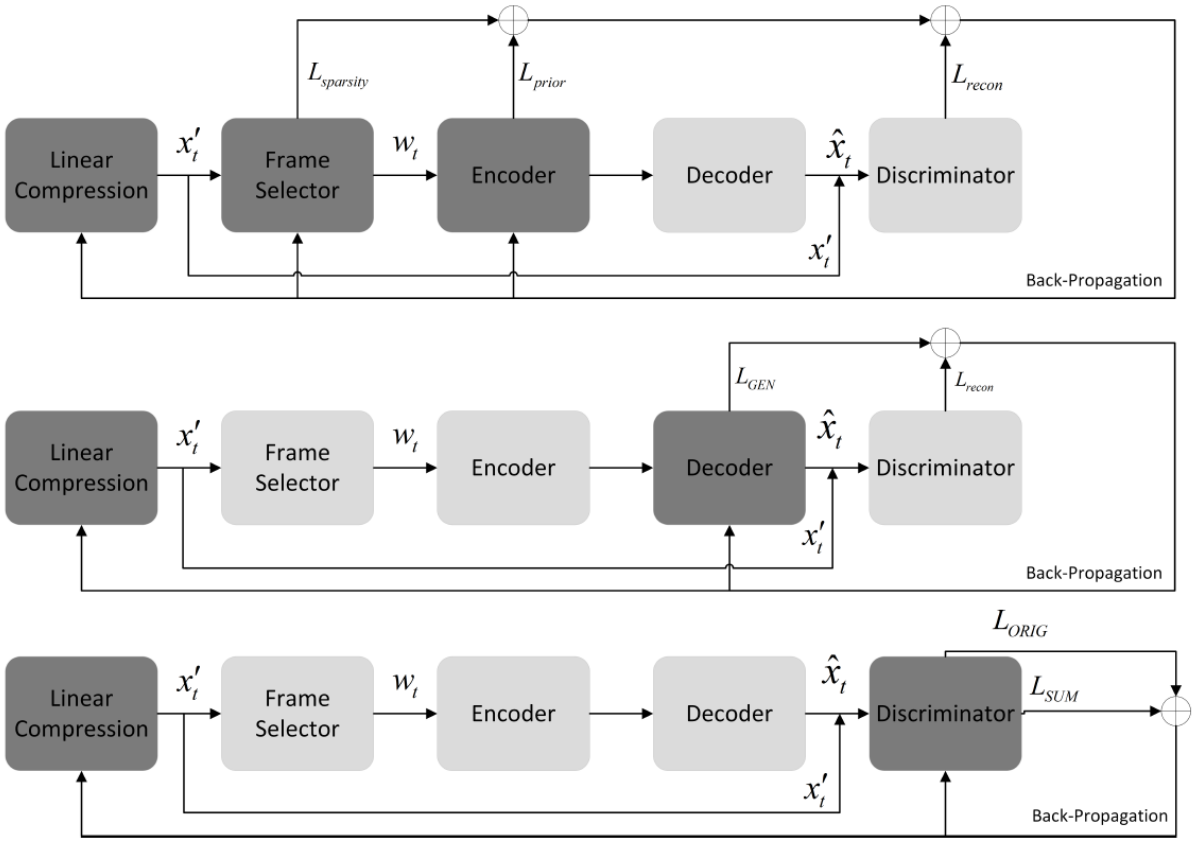
$$L_{GEN} = \left(1 - cLSTM\left(\hat{X}\right)\right)^2 \tag{17}$$

where L_{GEN} is used to minimise the MSE between the label of the original video and the probability computed when the discriminator processes video reconstructions based on the summary.

3.8 Model training

In our network model, the training process is primarily divided into three steps, as depicted in Figure 5. Specifically, for a video of T frames, x_t, x_t' and $\hat{x}_t$ represent the original, compressed and reconstructed feature vectors, respectively, where $t \in [1, T]$ denotes the frame index. During the training process, the parameters to be learned are θ_l, θ_s, θ_e, θ_d and θ_c, corresponding respectively to the linear compression layer, frame selector, encoder, decoder and discriminator.

Figure 5 The training process of TCN-SUM (see online version for colours)



In each step of training, the modules participating in the training are represented by dark modules.

In the first step of the training process, to learn parameter $\{\theta_l, \theta_s, \theta_e\}$, the algorithm performs the forward propagation of the model and minimises loss $L_{recon} + L_{sparsity} + L_{prior}$, followed by backward propagation to update the linear compression layer, frame selector and encoder.

In the second step of the training process, to learn parameter $\{\theta_l, \theta_d\}$, the algorithm executes the forward propagation of the model, computes loss $L_{recon} + L_{GEN}$, and then performs backward propagation to update the linear compression layer and the decoder.

In the third step of the training process, to learn parameter $\{\theta_l, \theta_c\}$, the algorithm performs the forward propagation of the model. Specifically, the compressed feature vectors of the original video $(x'_t, t \in [1, T])$ are passed through the discriminator, and loss L_{ORIG} is computed using a label-based approach. Similarly, the reconstructed feature vectors $(\hat{x}_t, t \in [1, T])$ are processed by the discriminator, and loss L_{SUM} is calculated. Then, loss $L_{ORIG} + L_{SUM}$ is minimised through backward propagation to update the linear compression layer and the discriminator.

4 Experiments

4.1 Data set

We primarily conducted experiments and analysed results using two data sets: SumMe and TVSum. The SumMe data set comprises 25 user videos depicting various events, including sports, holidays, cooking and more. These videos encompass scenarios with both rapid and gradual scene changes. Video durations range from 1 minute to 6.5 minutes. Each video comes with annotations, with most having 15 user annotations and a maximum of 18 user annotations. The TVSum data set contains 50 videos sourced from YouTube. These videos are selected from 10 categories in TRECVid MED, with each category having 5 videos. The video lengths span from 1.5 minutes to 11 minutes. Each video in TVSum is annotated by 20 users, using shot-level and frame-level importance scores. Further details about the data sets are presented in Table 2.

Table 2 Description of SumMe and TVSum

Name	*Number*	*Duration (min)*	*Number of labels*	*Type of label*
SumMe	25	1–6.5	15–18	Key shot and frame-level importance scores
TVSum	50	1.6–11	20	frame-level importance scores

4.2 Experimental metrics

The F1-score is commonly used to evaluate the performance of generated video summaries. Thus, we employ the F1-score to assess the model's performance. The specific evaluation process is as follows: the generated keyshot-based summary A is constrained to be within 15% of the original video duration; then, precision and recall are calculated against the Ground-Truth summary B based on the temporal overlap between them; finally, the F1-score is computed based on precision and recall.

Precision is a measure indicating the ratio of the common portion between the model-generated summary A and the Ground-Truth summary B to the generated summary. It is represented as:

$$P = \frac{A \cap B}{A} \tag{18}$$

Recall is a measure indicating the ratio of the common portion between the model-generated summary A and the Ground-Truth summary B to the Ground-Truth summary itself. It is represented as:

$$R = \frac{A \cap B}{B} \tag{19}$$

The F1-score is a balance between precision and recall, represented as:

$$F1 - score = \frac{2 \times P \times R}{P + R} \times 100\% \tag{20}$$

4.3 Implementation details

The computer configuration we used is as follows: CPU is Intel Core i9-11900KF, GPU is NVIDIA GeForce RTX 2080 Ti 11GB. In our experiments, a total of 100 epochs were set, training was conducted using the Adam optimiser, with a learning rate of 10^{-4} for all components except the discriminator, a learning rate of 10^{-5} for the discriminator and a regularisation factor α of 0.3. For evaluation, we followed the standard 5-fold cross-validation method (where 80% of the videos were used for training and the remaining 20% for testing). Finally, we implemented our approach in PyTorch 1.7.1.

4.4 Comparison algorithms

To ensure fair comparisons with other methods, we employed the F1-score as the evaluation metric. The compared algorithms are as follows:

1) *SUM-GAN (Mahasseni et al., 2017)*: SUM-GAN is one of the most representative unsupervised video summarisation methods. It's also the first work to combine an LSTM-based keyshot selector with a VAE and a trainable discriminator through adversarial learning to acquire video summarisation.

2) *DR-DSN (Zhou et al., 2018)*: DR-DSN is the first work to apply reinforcement learning to unsupervised video summarisation. It formulates video summarisation as a sequential decision process, developing a deep summary network to generate video summaries and introducing a novel diversity-representativeness reward function.

3) *ACGAN (He et al., 2019)*: ACGAN primarily introduces Conditional GAN into the field of unsupervised video summarisation. It is also the first work to introduce frame-level multi-head self-attention for video summarisation.

4) *Cycle-SUM (Yuan et al., 2019)*: Cycle-SUM primarily employs a novel cycle-consistent adversarial LSTM architecture that effectively maximises the mutual information between the summarised video and the original video.

5) *GL-RPE (Jung et al., 2020)*: GL-RPE introduces an input decomposition strategy aimed at capturing both local and global information from input video frames. This approach aims to overcome the challenges of LSTM's limited effectiveness in modeling long sequences.

6) *SUM-GAN-AAE (Apostolidis et al., 2020)*: SUM-GAN-AAE primarily introduces an attention-based VAE architecture. This method also serves as our baseline model.

7) *RSGN (Zhao et al., 2021)*: RSGN primarily introduces graph convolution into video summarisation. It proposes a Reconstruct Sequence Graph Network to encode frames and shots into hierarchical sequences and graphs. Frame-level dependencies are encoded using LSTM, while shot-level dependencies are captured by Graph Convolutional Networks. At the same time, the corresponding unsupervised version RSGNuns is given.

8) *SUM-FCN (Rochan et al., 2018)*: SUM-FCN treats the video summarisation task as semantic segmentation and proposes a fully convolutional sequence network. Its structure is based on convolutional and deconvolutional layers, and as the network deepens, the effective context size increases. This enables the network to effectively model long-term dependencies between frames and learn frame importance.

9) *iPTNet (Jiang and Mu, 2022)*: The introduction of iPTNet aims to address the issue of data scarcity in supervised video summarisation methods. To tackle this problem, the method combines video summarisation with video moment localisation and performs cross-sample task transfer. This approach supplements the training data set, leading to superior results.

4.5 Quantitative analysis

In this experiment, we compared our algorithm with various unsupervised methods, including SUM-GAN, DR-DSN, ACGAN, Cycle-SUM, GL-RPE, SUM-GAN-AAE, RSGN, as well as supervised methods like SUM-FCN and iPTNet. During the comparison process, we employed the F1-score as the evaluation metric, and the comparative results are presented in Table 3.

Table 3 F1-score comparison with contrasting algorithms

Method	*Supervised/ Unsupervised*	*SumMe*	*TVSum*
SUM-GAN (Mahasseni et al., 2017)	Unsupervised	39.1	51.7
DR-DSN (Zhou et al., 2018)	Unsupervised	41.4	57.6
ACGAN (He et al., 2019)	Unsupervised	46.0	58.5
Cycle-SUM (Yuan et al., 2019)	Unsupervised	41.9	57.6
GL-RPE (Jung et al., 2020)	Unsupervised	50.2	59.1
SUM-GAN-AAE (Apostolidis et al., 2020)	Unsupervised	48.9	58.3
RSGNuns (Zhao et al., 2021)	Unsupervised	42.3	58.0
RSGN (Zhao et al., 2021)	Supervised	45.0	60.1
SUM-FCN (Rochan et al., 2018)	Supervised	47.5	56.8
iPTNet (Jiang et al., 2022)	Supervised	54.5	63.4
Ours	Unsupervised	**50.4**	**60.5**

Firstly, comparing our method with the unsupervised approaches, it can be observed that our method achieves optimal results on the SumMe and TVSum data sets, with F1-scores of 50.4 and 60.5, respectively. In contrast to LSTM-based methods such as SUM-GAN (F1-scores of 39.1 and 51.7) and Cycle-SUM (F1-scores of 41.9 and 57.6), our method demonstrates a significant performance improvement. This validates the advantages of Bi-TCN in handling temporal modeling, especially for videos characterised by long sequences. Compared to GL-RPE (F1-scores of 50.2 and 59.1), our method also holds a slight lead. Moreover, the end-to-end training process of our approach is more straightforward compared to the need for video segmentation in GL-RPE, highlighting the advantages of our Bi-TCN over LSTM-based methods.

Furthermore, when comparing our method to supervised approaches, it becomes evident that our approach outperforms supervised method like SUM-FCN (F1-scores of 47.5 and 56.8). While our method's summarisation performance falls slightly behind iPTNet (F1-scores of 54.5 and 63.4), this discrepancy can be largely attributed to the fact that iPTNet employs joint training to gather extensive training data, which significantly surpasses the amount of data available to our approach.

Finally, when comparing our method to our baseline approach SUM-GAN-AAE (F1-scores of 48.9 and 58.3), it becomes evident that our method outperforms the baseline method by a significant margin on both SumMe and TVSum data sets. This observation underscores the superiority of our proposed frame selector module based on TCN in contrast to the baseline model's LSTM-based frame selector module. Our method effectively establishes long-term dependencies between video frames, thus emphasising the importance of key frames more effectively.

4.6 Ablation study

To verify that Bi-TCN can effectively establish long-term dependencies between video frames, we conducted an ablation experiment on the frame selection module to provide substantial evidence. Specifically, we excluded the self-attention mechanism from the frame selection module, utilising only the Bi-TCN module to establish inter-frame dependencies. This variant is named TCN-SUMw/o_sa. The results are presented in Table 4.

Table 4 Ablation results study

Method	*SumMe*	*TVSum*
SUM-GAN-AAE (et al.2020)	48.9	58.3
TCN-SUM	**50.4**	**60.5**
TCN-SUMw/o_sa	49.7	59.6

By comparing SUM-GAN-AAE and TCN-SUMw/o_sa, it is evident that TCN-SUMw/o_sa outperforms our baseline model SUM-GAN-AAE. The F1-scores achieved by TCN-SUMw/o_sa are 49.7 and 59.6, while those for SUM-GAN-AAE are 48.9 and 58.3, respectively. This improvement of 0.8 and 1.3 in F1-scores indicates that our Bi-TCN is superior to Bi-LSTM in establishing long-term dependencies between frames in SumMe and TVSum. Since the video reconstructor and discriminator modules are the same in both methods, and only Bi-TCN or Bi-LSTM is used in the frame selection, we can conclude that Bi-TCN may achieve better results than Bi-LSTM in video summarisation tasks. Additionally, by comparing the results of TCN-SUM and TCN-SUMw/o_sa, it is evident that TCN-SUM performs significantly better than TCN-SUMw/o_sa, with F1-scores of 50.4 and 60.5 compared to 49.7 and 59.6, respectively. This outcome indicates that the self-attention mechanism effectively highlights important parts of the video, emphasising the necessity of self-attention for video summarisation.

Additionally, we conducted ablation experiments on the regularisation factor σ. We evaluated several values, specifically σ=0.15, σ=0.3, σ=0.4 and σ=0.5. However, we observed that when σ exceeded 0.5, there was a noticeable decline in performance on the test set. Consequently, we refrained from conducting ablation experiments for $\sigma > 0.5$. The outcomes of these ablation experiments on the regularisation factor are presented in Table 5. Analysing the results reveals that the effectiveness of σ doesn't strictly follow a 'bigger is better' pattern; instead, it appears to exhibit a trend of an initial increase followed by a subsequent decrease in performance. Particularly noteworthy is the observation that the optimal performance was achieved when $\sigma = 0.3$ was set as the value for the regularisation factor. Even at σ=0.15, our model maintained a substantial competitive edge in terms of performance.

Table 5 Study on ablation results of different regularisation factors

	SumMe	*TVSum*
σ=0.15	49.4	59.4
σ=0.3	**50.4**	**60.5**
σ=0.4	47.9	59.1
σ=0.5	46.3	57.7

4.7 Qualitative analysis

For a more intuitive analysis of the quality of the summaries generated by our method, we also provide a visual analysis of the results. Specifically, we employed KeyShot Technique (KTS) for shot segmentation and visualised the last frame of each shot to vividly depict the quality of the generated summaries. The visual analysis results of TCN-SUM are presented in Figure 6, where we showcase the visualisation results of video 15 from the TVSum data set. This video demonstrates the process of cleaning a dog's ears. In the visualised results, the gray segments represent frame importance scores, while the brown segments indicate selected key frames.

Through Figure 6, it becomes evident that our method accurately captures the significant portions of the video. The visualisation results offer a clear understanding of the process of cleaning a dog's ears: First, the cleaning solution is prepared; Second, an assistant holds the dog in place; Third, the dog's ears are lifted; Fourth, the cleaning solution is sprayed and finally, cotton balls are used to clean the dog's ears. This visualisation provides a clear insight into the main steps of the video content, showcasing the effectiveness of our TCN-SUM method.

To investigate the importance of the attention mechanism in our method, we also conducted a visual analysis of the results obtained from TCN-SUMw/o_sa. We employed the same video 15 from the TVSum data set for this analysis, and the visualisation results are presented in Figure 7. In Figure 7, although TCN-SUMw/o_sa is still capable of capturing the significant parts of the video, we cannot deduce the specific process of cleaning a dog's ears from the generated summary. This observation underscores the critical role of the attention mechanism in video summarisation. The absence of the attention mechanism in TCN-SUMw/o_sa results in a less informative summary that fails to convey the step-by-step process of cleaning a dog's ears. This further demonstrates that the attention mechanism plays a crucial role in enhancing the quality and interpretability of video summaries.

Figure 6 Visualisation results of TCN-SUM (see online version for colours)

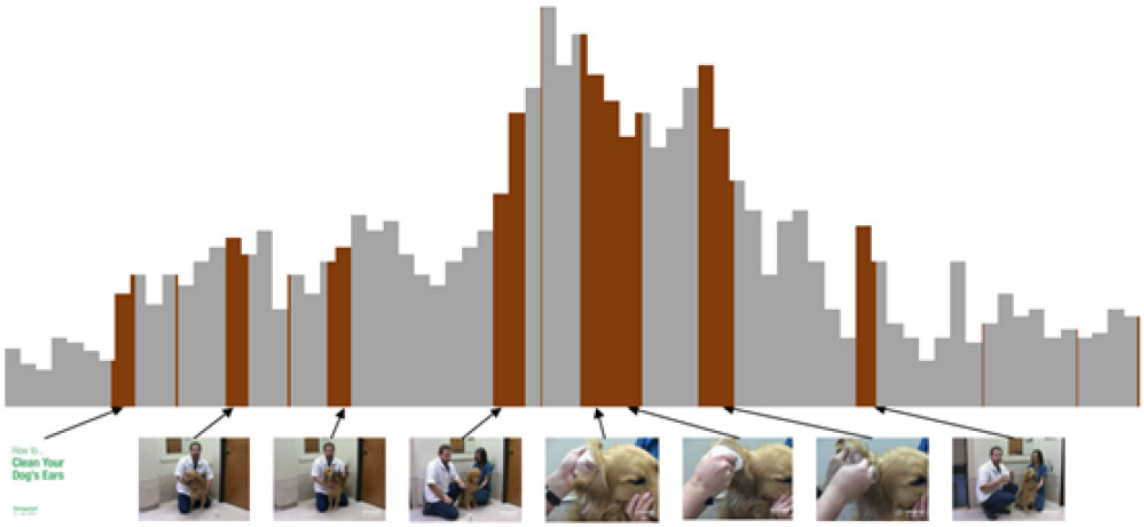
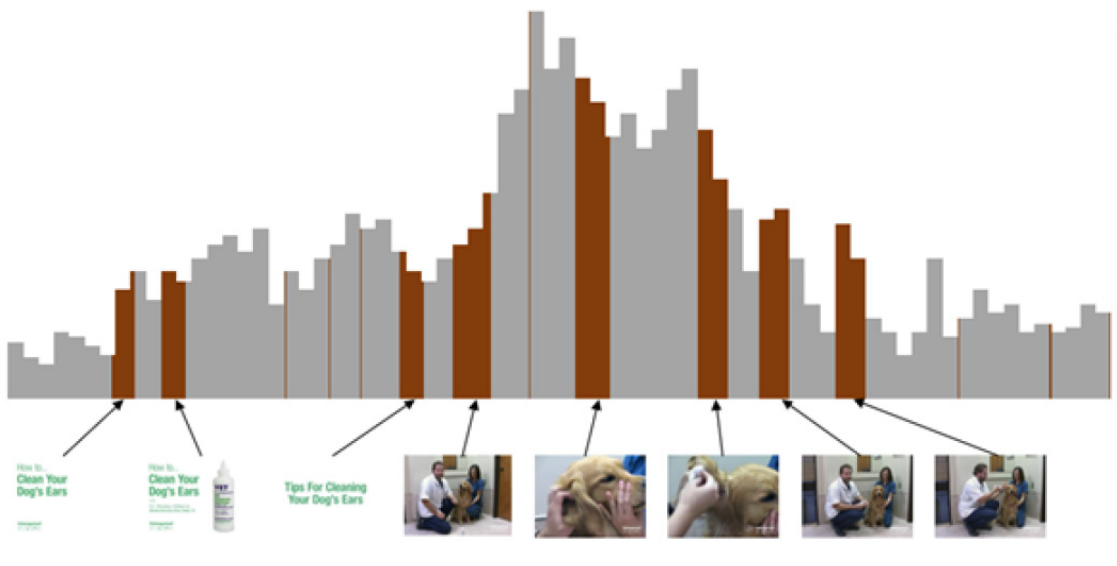


Figure 7 Visualisation results of TCN-SUMw/o_sa (see online version for colours)



5 Conclusion

This paper introduces an unsupervised video summarisation model based on Bi-TCN combined with a self-attention mechanism to select a subset of key frames. Bi-TCN effectively models the contextual relationships within videos, utilising residual connections to preserve temporal sequencing. The self-attention mechanism further complements the inter-frame dependencies. We also employ an attention-based generator model to reconstruct videos that closely resemble the original content. Through extensive experiments, we demonstrate the effectiveness of our approach, achieving performance improvements of 1–10% over previous state-of-the-art methods and maintaining competitiveness when compared to recent supervised methods. Additionally, qualitative visualisations are provided to illustrate how our method successfully emphasises important segments within videos without requiring any supervision.

Acknowledgements

This work is supported by the Dalian Excellent Young Project (No. 2022RY35), National Natural Science Foundation of China (No. 61902050), Natural Science Foundation of Liaoning Province (No. 2021-MS-136) and the Fundamental Research Funds for the Central Universities (No. 3132019355).

References

Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V. and Patras, I. (2020) 'Unsupervised video summarization via attention-driven adversarial learning', *MultiMedia Modeling: 26th International Conference*, pp.492–504.

Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V. and Patras, I. (2021) 'AC-SUM-GAN: connecting actor-critic and generative adversarial networks for unsupervised video summarization', *IEEE Transactions on Circuits and Systems for Video Technology*, pp.3278–3292.

Apostolidis, E., Balaouras, G., Mezaris, V. and Patras, I. (2022) 'Summarizing videos using concentrated attention and considering the uniqueness and diversity of the video frames', *Proceedings of the International Conference on Multimedia Retrieval*, pp.407–415.

Bai, S., Kolter, J.Z. and Koltun, V. (2018) 'An empirical evaluation of generic convolutional and recurrent networks for sequence modeling', *arXiv:1803.01271*.

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

Chen, Y., Guo, B., Shen, Y., Zhou, R., Lu, W., Wang, W. and Suo, X. (2022) 'Video summarization with u-shaped transformer', *Applied Intelligence*, Vol. 52, No. 15, pp.17864–17880.

Fajtl, J., Sokeh, H.S., Argyriou, V., Monekosso, D. and Remagnino, P. (2019) 'Summarizing videos with attention', *Asian Conference on Computer Vision Workshops*, pp.39–54.

Feng, S., Xie, Y., Wei, Y., Yan, J. and Wang, Q. (2022) 'Transformer-based video summarization with spatial-temporal representation', *Proceedings of the 8th International Conference on Big Data and Information Analytics*, pp.428–433.

Fu, T-J., Tai, S-H. and Chen, H-T. (2019) 'Attentive and adversarial learning for video summarization', *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp.1579–1587.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014) 'Generative adversarial nets', *Annual Conference on Neural Information Processing Systems*, pp.2672–2680.

Graves, A. and Schmidhuber, J. (2005) 'Framewise phoneme classification with bidirectional LSTM networks', *Neural Networks*, pp.602–610.

He, X., Hua, Y., Song, T., Zhang, Z., Xue, Z., Ma, R., Robertson, N. and Guan, H. (2019) 'Unsupervised video summarization with attentive conditional generative adversarial networks', *Proceedings of the 27th ACM International Conference on Multimedia*, pp.2296–2304.

Ho, H-I., Chiu, W-C. and Wang, Y-C.F. (2018) 'Summarizing first-person videos from third persons' points of views', *Proceedings of the European Conference on Computer Vision*, pp.70–85.

Hsu, T-C., Liao, Y-S. and Huang, C-R. (2023) 'Video summarization with spatiotemporal vision transformer', *IEEE Transactions on Image Processing*, Vol. 32, pp.3013–3026.

Hu, X., Hu, X., Li, J. and You, K. (2023) 'Generative adversarial networks for video summarization based on key-frame selection', *Information Technology and Control*, Vol. 52, No. 1, pp.185–198.

Ji, Z., Xiong, K., Pang, Y. and Li, X. (2020) 'Video summarization with attention-based encoder–decoder networks', *IEEE Transactions on Circuits and Systems for Video Technology*, pp.1709–1717.

Jiang, H. and Mu, Y. (2022) 'Joint video summarization and moment localization by cross-task sample transfer', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.16388–16398.

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

Jung, Y., Cho, D., Woo, S. and Kweon, I.S. (2020) 'Global-and-local relative position embedding for unsupervised video summarization', *Proceedings of the European Conference on Computer Vision*, pp.167–183.

Kingma, D.P. and Welling, M. (2014) 'Auto-encoding variational bayes', *International Conference on Learning Representations*, pp.14–28.

Kulesza, A. and Taskar, B. (2012) 'Determinantal point processes for machine learning', *Machine Learning*, pp.123–286.

Mahasseni, B., Lam, M. and Todorovic, S. (2017) 'Unsupervised video summarization with adversarial LSTM networks', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.202–211.

Meena, P., Kumar, H. and Yadav, S.K. (2023) 'A review on video summarization techniques', *Engineering Applications of Artificial Intelligence*, Vol. 118, pp.105667–105667.

Mirza, M. and Osindero, S. (2014) 'Conditional generative adversarial nets', *Computer Science*, pp.2672–2680.

Panda, R., Das, A., Wu, Z., Ernst, J. and Roy-Chowdhury, A.K. (2017) 'Weakly supervised summarization of web videos', *Proceedings of the IEEE International Conference on Computer Vision*, pp.3657–3666.

Rochan, M. and Wang, Y. (2019) 'Video summarization by learning from unpaired data', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.7902–7911.

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

Sabha, A. and Selwal, A. (2023) 'CoSumNet: a video summarization-based framework for COVID-19 monitoring in crowded scenes', *Artificial Intelligence in Medicine*, Vol. 139, pp.102544–102544.

Sreeja, M.U. and Kovoor, B.C. (2023) 'A multi-stage deep adversarial network for video summarization with knowledge distillation', *Journal of Ambient Intelligence and Humanized Computing*, Vol. 14, No. 8, pp.9823–9838.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017) 'Attention is all you need', *Annual Conference on Neural Information Processing Systems*, pp.5998–6008.

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

Yuan, Y. and Zhang, J. (2023) 'Unsupervised video summarization via deep reinforcement learning with shot-level semantics', *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 33, No. 1, pp.445–456.

Zhang, K., Chao, W-L., Sha, F. and Grauman, K. (2016) 'Video summarization with long short-term memory', *Proceedings of the European Conference on Computer Vision*, pp.339–348.

Zhao, B., Gong, M. and Li, X. (2022) 'Hierarchical multimodal transformer to summarize videos', *Neurocomputing*, Vol. 468, pp.360–369.

Zhao, B., Li, H., Lu, X. and Li, X. (2021) 'Reconstructive sequence-graph network for video summarization', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.2793–2801.

Zhao, B., Li, X. and Lu, X. (2018) 'Hsa-rnn: hierarchical structure-adaptive rnn for video summarization', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.7405–7414.

Zhou, K., Qiao, Y. and Xiang, T. (2018) 'Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward', *Proceedings of the AAAI Conference on Artificial Intelligence*, pp.7582–7589.

Zhu, J-Y., Park, T., Isola, P. and Efros, A.A. (2017) 'Unpaired image-to-image translation using cycle-consistent adversarial networks', *Proceedings of the IEEE International Conference on Computer Vision*, pp.2223–2232.