Neurocomputing 617 (2025) 128945

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Graph convolutional network for fast video summarization in compressed domain

Chia-Hung Yeh [a,b], Chih-Ming Lien [a], Zhi-Xiang Zhan [b], Feng-Hsu Tsai [c], Mei-Juan Chen [d,*]

[a] Department of Electrical Engineering, National Taiwan Normal University, Taipei 106308, Taiwan
[b] Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 804201, Taiwan
[c] Advanced Streaming Technology Division, Platform and Growth Center, KKCompany, Taipei 115020, Taiwan
[d] Department of Electrical Engineering, National Dong Hwa University, Hualien 974301, Taiwan

ARTICLE INFO

Communicated by Zidong Wang

Keywords:
Video Summarization
Video Compression
Graph Convolutional Network
Compressed Domain

ABSTRACT

Video summarization is the process of generating a concise and representative summary of a video by selecting its most important frames. It plays a vital role in the video streaming industry, allowing users to quickly understand the overall content of a video without watching it in its entirety. Most existing video summarization methods require fully decoding the video stream and extracting the features with a pre-trained deep learning model in the pixel domain, which is time-consuming and computationally expensive. To address this issue, this paper proposes a novel method called Graph Convolutional Network-based Compressed-domain Video Summarization (GCNCVS), which directly exploits the compressed-domain information and leverages graph convolutional network to learn temporal relationships between frames, thereby enhancing its ability to capture contextual and valuable information when generating summarized videos. To evaluate the performance of GCNCVS, we conduct experiments on two benchmark datasets, SumMe and TVSum. Experimental results demonstrate that our method outperforms existing methods, achieving an average F-score of 53.5% on the SumMe dataset and 72.3% on the TVSum dataset. Additionally, the proposed method shows Kendall's τ correlation coefficient of 0.157 and Spearman's ρ correlation coefficient of 0.205 on the TVSum dataset. Our method also significantly reduces computational time, which enhances the feasibility of video summarization in video streaming environments.

1. Introduction

The demand for streaming video from platforms such as YouTube, Netflix, and Disney+ is constantly increasing, resulting in a significant surge in the volume of video data. High-quality video with characteristics such as wide color gamut, ultra-high definition (UHD), and high dynamic range (HDR) is gaining popularity. Therefore, video compression must be used to overcome storage and network limitations for a better streaming experience.

Video summarization involves extracting keyframes or segments from a video to create a shorter version that captures the most relevant content. It is a key technology in the video streaming industry because it allows users to quickly grasp the general content of the video. Video summarization has a wide range of applications, from surveillance video analysis [1] to medical video browsing [2]. However, the task of quickly summarizing high-quality videos poses a formidable obstacle due to the complexity and richness of the information contained in videos.

Most video summarization methods can be divided into three main steps: feature extraction, frame-level importance score prediction, and keyshot selection. According to the type of data used, video summarization methods can be further divided into two categories: pixel-domain methods and compressed-domain methods. With the rapid advancement of neural networks [3–5] and deep learning (DL), DL-based methods have achieved significant performance improvements in video summarization tasks. However, most deep learning-based methods use pixel-domain information by first decoding the video to extract features using a pre-trained deep learning network, such as GoogLeNet [6] and ResNet [7]. This process demands significant computational resources, making it time-consuming and computationally expensive. In contrast, compressed-domain methods offer significant advantages by utilizing information directly from compressed video without the necessity of fully decoding and applying a pre-trained model to extract features from

* Corresponding author.
E-mail address: cmj@gms.ndhu.edu.tw (M.-J. Chen).

https://doi.org/10.1016/j.neucom.2024.128945
Received 26 March 2024; Received in revised form 18 September 2024; Accepted 13 November 2024
Available online 20 November 2024

the pixel domain, resulting in computational savings. Over the past three decades, many video coding standards have been developed [8]. The basic idea of compression is to preserve meaningful information by reducing redundancy, which enables encoders to act as efficient feature extractors. Compressed-domain information thus represents various meaningful features such as DC coefficients for pattern recognition, motion vectors (MV) to indicate the degree of object motion, and quantization parameters (QP) to indicate frame quality. In brief, leveraging compressed-domain information for video summarization offers several advantages including improving efficiency by retaining essential data while discarding redundancies, reducing computational load by minimizing the need to process raw video, and improving performance by focusing on the most relevant content.

Identifying useful features from compressed data is often difficult because each compression technique poses additional constraints such as nonlinear processing and resolution reduction. Furthermore, compressed-domain features are embedded within data exhibiting a non-Euclidean structure. Traditional convolutional neural networks are ill-suited to handle non-Euclidean structured data because discrete convolution fails to maintain translation invariance on non-Euclidean structured data. Furthermore, capturing temporal relationships is crucial for video summarization. Many existing video summarization methods use recurrent neural networks (RNNs) to capture temporal relationships between video frames. Zhou et al. [9] utilized a reinforcement learning framework to train a long short-term memory (LSTM) network with a diversity-representation reward to motivate the model to generate diverse and representative video summaries. However, RNN-based approaches face challenges, including the long distances over which the forward and backward signals must propagate within the network [10]. This limitation negatively impacts the network's ability to capture global relationships, which is also critical for video summarization [11]. Moreover, RNN-based methods encounter challenges in realizing parallel operations during the training [12]. Attention-based methods are proposed to address these problems and have shown impressive performance in video summarization. Li et al. [12] designed a global diverse attention to investigate the pairwise temporal relationships of video frames for constructing the video summary. Hsu et al. [13] proposed a spatiotemporal vision transformer to explore the temporal relationships before identifying the spatial features to classify the importance of frames. Khan et al. [14] developed a deep multi-scale pyramidal features network based on a vision transformer to extract and refine the representation features of video frames for generating the video summary. Although these approaches achieve satisfactory results, they process the pixel-domain information rather than the compressed-domain information readily available in modern videos. In this paper, the goal is to efficiently exploit the valuable information available in the compressed-domain information while considering temporal relationships among video frames, without compromising the model's ability to generalize across diverse videos. We propose a methodology to transform compressed-domain features into data topology and efficiently extract features for learning. Graph convolutional network (GCN) [15] can establish topological associations for compressed-domain data in normed spaces and extract features of topological graphs by exploiting the relationships between connected nodes. The proposed method utilizes the GCN and learns directly from compressed videos, which leads to substantial computational savings by avoiding full decoding of video stream, thereby greatly accelerating the processing of video summarization.

In our proposed method, we first obtain compressed-domain information, which includes macroblock and sub-macroblock partition type, motion vector, residual, DC coefficient, and quantization parameter. Each of these individually builds a graph and GCN is used to learn the temporal relationships between frames. The frames with compressed-domain information are treated as nodes, and connections between nodes are established based on the evaluation of their similar characteristics to form a graph. Subsequently, we employ the multi-layer perceptron (MLP) to fuse the features, which enables the model to generate features that carry the most critical information. By incorporating GCN, we further study the relationships between frames based on the important information generated through the feature fusion process. This allows the model to gain a deeper understanding of potential connections among video frames. Finally, our model utilizes the MLP to predict frame importance. By evaluating the importance of each frame, we can select keyframes that contain the most important aspects of the video. These keyframes are used to generate video summaries. Our primary contributions are as follows:

(1) To the best of our knowledge, this paper presents a pioneering effort to exploit multiple types of compressed-domain information in the GCN-based deep learning method for video summarization.
(2) We leverage the graph convolutional network to efficiently address the non-Euclidean structure inherent in compressed-domain information and capture temporal relationships in video.
(3) Comprehensive experiments show that GCNCVS achieves an average F-score of 53.5% on the SumMe dataset and 72.3% on the TVSum dataset. Additionally, the proposed method shows Kendall's τ correlation coefficient of 0.157 and Spearman's ρ correlation coefficient of 0.205 on the TVSum dataset. It avoids the time for full decoding of the video stream and extracting the features with the pre-trained model, resulting in dramatically reduced computational time compared to other prevailing methods, making it more feasible for video summarization in video streaming environments.

The remainder of this paper is divided into four sections. Section 2 describes the related video summarization methods. Section 3 details the proposed GCNCVS approach. Experimental results and performance comparison are detailed in Section 4. Finally, Section 5 outlines a conclusion.

2. Related work

In this section, we provide a concise overview of the related video summarization methods in both the pixel and the compressed domains.

2.1. Pixel domain

Numerous studies have been conducted to summarize videos in the pixel domain [16]. Zhang et al. [17] utilized an LSTM to model the temporal dependencies between video frames and estimate the importance scores of frames. Mahasseni et al. [18] proposed an adversarial LSTM network with a variational autoencoder and discriminator for generating a summary-based reconstructed video and discriminating the summary-based reconstructed video from the original video, respectively. Jung et al. [19] designed a two-stream model that captures local and global temporal information while incorporating variance loss to encourage the model to generate different importance scores for each frame. Kashid et al. [20] used a self-organizing map clustering on the spatial-temporal features extracted from modified GoogLeNet, ResNet, and LSTM to form the video summary.

Fajtl et al. [21] used a self-attention mechanism to predict the importance scores of frames. Apostolidis et al. [22] divided the video into a fixed number of non-overlapping segments and learned the frame relationships within each segment before addressing the relationships between segments. Zhu et al. [23] considered video summarization as a process of temporal interest detection. The method predicts both the segment locations and corresponding importance scores. Zhu et al. [24] partitioned video into equal lengths and applied multiscale hierarchical attention to learn short-range and long-range temporal representations to predict the importance of each frame. Liang et al. [25] designed a convolutional-attention generator for predicting the importance scores

of frames and generating weighted frame features. In addition, they developed an LSTM-based discriminator to discriminate between weighted frame features and original features. Liang et al. [26] proposed a dual-path attentive framework that integrates appearance features with importance scores to learn global dependencies and continuously refines previous frame important scores. Zhao et al. [27] proposed a hierarchical transformer to integrate the visual and audio features to capture the scene information for shots and select the most representative shots for the video summary. Li et al. [28] proposed a transformer to jointly model the semantic dependencies across videos and connect them with video summarization. Wang et al. [29] designed a reinforcement learning structure with two policies for sampling keyframes from video and refining the sampled keyframes. Argaw et al. [30] developed a method by autoregressively decoding the multimodal features containing textual and visual features to produce a video summary. Huang et al. [31] designed an aesthetic encoder to capture the aesthetic attributes in the video and fuse the visual, aesthetic, textual, and audio information with an attention-based multimodal module to highlight the most important shots for the video summary. Huang et al. [32] utilized causal learning technique and Bayesian probability to understand the causal relationships in the video for generating the video summary.

Wu et al. [33] pioneered the use of GCN in multi-video summarization, introducing bagging and penalty loss strategies to balance data distribution and penalize model misclassification of minority classes. Additionally, their proposed method [34] improves the previous work [33] by replacing static GCN with dynamic GCN. Park et al. [35] considered the frame as a node in a graph and used GCN to recursively refine the graph and predict the importance score for each node. Zhao et al. [36] used a bidirectional LSTM to extract the frame-level temporal dependencies within video shots and applied a GCN to estimate the dependencies between different video shots. The video summary is generated by leveraging pairwise dependencies among video shots. Zhang et al. [37] used reinforcement and contrastive learning to learn informative and discriminative shot-level features and designed a dissimilarity-guided attention graph to capture local and global information across shots. Zhu et al. [38] proposed to apply a GCN to learn the relationships between objects within a frame and then model the relationships between different frames. Zhong et al. [39] proposed graph information bottleneck-based semantic representation and attention alignment-based contextual feature transformation to integrate visual and semantic features into higher-level features for generating the video summary. Wu et al. [40] utilized the spatial-temporal graph convolutional network to extract the spatial and temporal relationships in video and create the video summary. As discussed earlier, many methods require full decoding of the video stream and applying a pre-trained deep learning model to extract the features in the pixel domain, which is prolonged and costly in computation.

2.2. Compressed domain

Several video summarization methods in the compressed domain have been studied [41]. Chew et al. [42] applied a clustering method to extract the representative frames based on DC histograms. Yu *et al.* [43] used a content-based adaptive clustering algorithm that groups the video frames based on motion vectors and DC histograms to generate the summary. Ren et al. [44] classified the level for each frame based on motion activities and human objects. The video summary is generated by selecting the higher-level frames. Wang and Ngo [45] used a hierarchical hidden Markov model of input motion features to generate the object-based and event-based rushes video summary. Dong et al. [46] selected candidate keyframes according to the statistical characteristics of compressed-domain information, including motion vectors, transform coefficients, prediction modes, and macroblock bit consumption. Subsequently, they pruned undesirable candidate frames according to the pixel-domain information of I frames to complete the summarization of H.264/AVC compressed video. Lakshya et al. [47] proposed a method to feed MV information into modified DSNet [23] for video summarization. Their modification focuses on the feature extractor module in [23], aiming to bridge the data distribution gap between the pixel data and MV data. This is achieved by applying the strategy of masked features to multi-head attention and introducing an aggregation sub-module within the feature extractor module. As previously reviewed, the research area of deep learning-based video summarization methods in the compressed domain remains relatively unexplored, with limited recent work targeting this area.

3. Proposed method

In this section, we provide the details of our proposed method consisting of the selection of compressed-domain information, the problem formulation, the details of the proposed framework, and the learning strategy.

3.1. Selection of compressed-domain Information

Accurate selection of representative features is crucial for successful video summarization. We choose five features, including macroblock and sub-macroblock partition type (MPT), motion vector (MV), residual (Res), DC coefficient (DC), and quantization parameter (QP) as the input of GCNCVS, and explain their meaning as follows.

According to the H.264/AVC standard [48], the input frame is divided into non-overlapping multiple macroblocks to facilitate motion estimation and compensation. The partition types for macroblocks support block sizes of 16×16, 16×8, 8×16 and 8×8. Furthermore, if the block size is selected as 8×8, it can be further divided into 8×4, 4×8, or 4×4. These different block sizes may reflect the texture of the video.

MV is a key parameter in video compression because it indicates the displacement of the best-matching block relative to the current encoding block. MV is derived from the motion estimation process, which identifies the block with the smallest motion cost in the reference frame. The motion estimation process can employ various search algorithms to obtain precise MVs, which convey valuable information about the motion characteristics of the video; the larger the MV, the faster the video content changes. As an important part of motion-compensated video compression, MV can also represent the motion information of the video summary.

The residual is a major aspect of video coding, as it may carry high-frequency information from high-motion regions. This information reveals the prediction error between the original frame and the motion-compensated frame. Transform coefficients include DC and AC coefficients, representing different characteristics in the residual. DC coefficients play a critical role in representing the average brightness or intensity of a block, serving as a key component for both intra-frame compression, where they capture the local spatial characteristics within a frame, and inter-frame compression, where they are utilized to encode the residual information between consecutive frames. The transform coefficients are then quantized to reduce redundancy. This step eliminates most of the AC coefficients since they contain visually trivial information. Therefore, only the DC coefficients containing the most essential information are selected in the proposed method. QP is also a key factor in video encoding; it determines the degree of distortion in compressed video. The higher the QP value, the larger the loss of video quality, but the bitrate also reduces. Therefore, the QP value is crucial for video summarization because it directly affects the quality of compressed video. Overall, MPT, MV, Res, DC, and QP are chosen as the inputs for GCNCVS. Each of these contributes essential and informative representations that enhance the performance of GCNCVS.

3.2. Problem formulation

Given a video stream, partially decode it to obtain compressed-

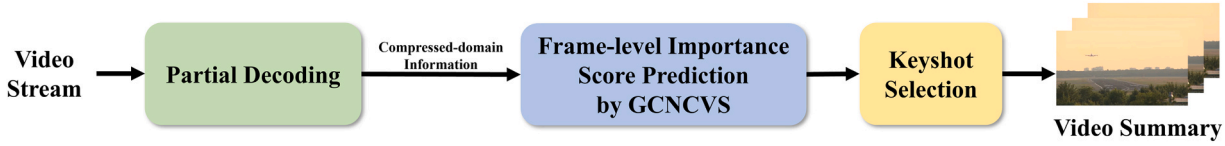

Fig. 1. Overview of the proposed framework. Given an input video stream, the video stream is first partially decoded to obtain the compressed-domain information. These features are used to predict frame-level importance scores through GCNCVS. The frame-level importance scores are subsequently derived into shot-level importance scores, from which keyshots are selected to form the video summary.

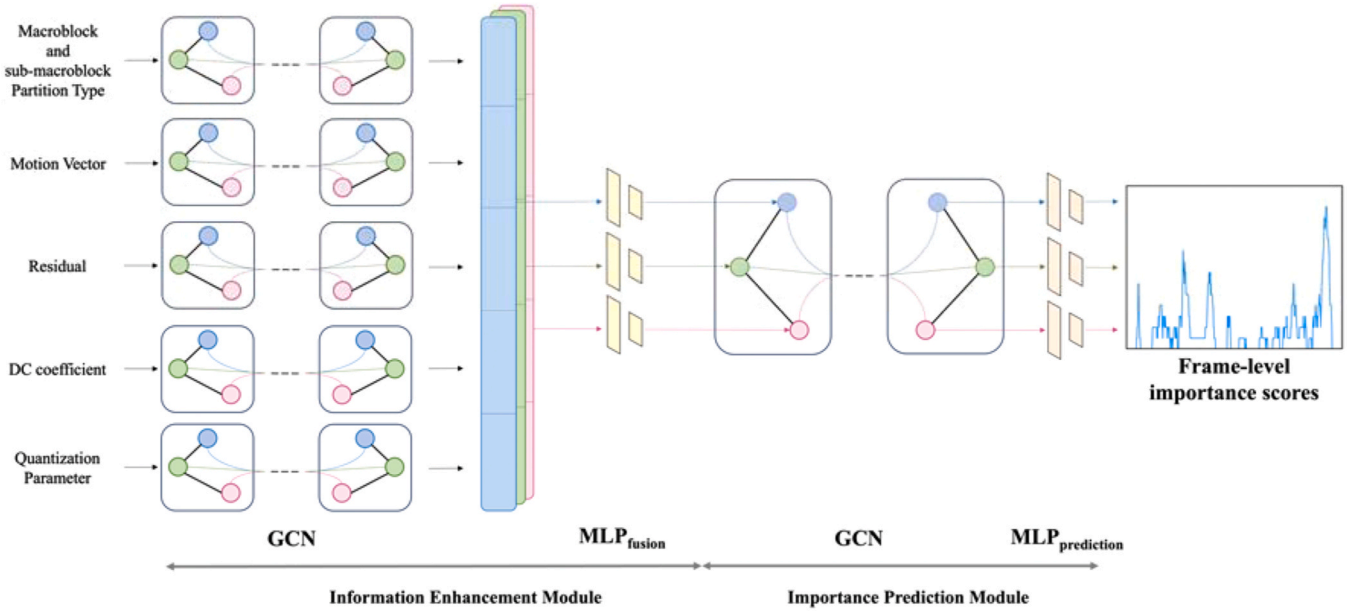

Fig. 2. The proposed GCNCVS framework. It consists of the information enhancement module and the importance prediction module. The information enhancement module enhances the input graphs through five parallel streams, each utilizing a GCN to handle an individual feature. These features are concatenated and then fused using the MLP to produce enhanced representations. The refined features are subsequently passed into the importance prediction module, which employs a GCN followed by the MLP to predict the importance score for each video frame.

domain information $X^c = \{x_n^c\}_{n=1}^N, c \in \{MBT, MV, Res, DC, QP\}$, where x_n^c represents the feature vectors associated with n^{th} frame, and N denotes the total number of frames. The objective is to apply the proposed method to generate a sequence $B = \{b_u\}_{u=1}^U$, where U denotes the total number of shots and $b_u \in \{0, 1\}$ indicates whether the u^{th} shot is selected as a keyshot for the summary. Specifically, the shot with $b_u = 1$ is chosen as a keyshot and included in the summary.

3.3. Proposed framework

Fig. 1 presents an overview of the proposed framework, which consists of the following three stages: video stream partial decoding, frame-level importance score prediction, and keyshot selection. Starting with the input video stream, the video stream is partially decoded to obtain the compressed-domain information. The information is processed using GCNCVS to estimate importance scores for each frame. The frame-level importance scores are combined to derive shot-level importance scores, which are used to identify and select keyshots for constructing the video summary.

3.3.1. Frame-level importance score prediction

Fig. 2 shows the framework of the proposed GCNCVS method, which consists of the information enhancement module and the importance prediction module. Details are discussed below.

The compressed-domain information X^c including *MPT*, *MV*, *Res*, *DC*, and *QP* is obtained to initialize the graphs after partially decoding the video stream. Each X^c is transformed into a corresponding graph $\mathcal{G}^c = (\mathcal{V}^c, \mathcal{E}^c)$. The node representation $\mathcal{V}^c = \{v_n^c\}_{n=1}^N$ is the set of feature vectors, where the n^{th} node corresponds to the n^{th} frame in the video, N is the total number of frames in the video, $v_n^c \in \mathbb{R}^{d_c}$ and d_c is the dimension of feature vectors. The edges $(v_n^c, v_m^c) \in \mathcal{E}^c$ represent the connections between nodes v_n^c and v_m^c for $n, m \in \{1, 2, \ldots, N\}$. The connections between nodes are established using the K-Nearest Neighbor (KNN) approach [49], which evaluates the cosine similarity *CS* between nodes and connects the K most similar neighbors for each node. The cosine similarity *CS* is computed as:

$$CS(v_n^c, v_m^c) = \frac{v_n^c \bullet v_m^c}{\|v_n^c\| \bullet \|v_m^c\|} \tag{1}$$

The information enhancement module enhances each type of information through five parallel streams, each consisting of two graph convolutional layers. The outputs features from the graph convolutional layers are concatenated and followed by the MLP for feature fusion. The importance prediction module includes two graph convolutional layers and the MLP for frame importance score prediction. All the graph convolutional layers in both modules follow the layer-wise propagation rule [15] and are shown as:

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

where $\widetilde{A}$ is the adjacency matrix, $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$ is a degree matrix derived

from $\widetilde{A}$, $\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$ is the normalization of adjacency matrix $\widetilde{A}$, $W^{(l)}$ is a trainable weight matrix for the l^{th} layer, $\sigma(\bullet)$ denotes an activation function, $H^{(l)}$ is the matrix of activations in the l^{th} layer, and $H^{(0)}$ denotes the input matrix of node feature vectors. Note that $\widetilde{D}^{-\frac{1}{2}}$ applies both row-wise and column-wise normalization to $\widetilde{A}$.

In the information enhancement module, the relationships among frames are strengthened by employing two graph convolutional layers for each type of compressed-domain information. After processing through the five parallel streams, the five output features are concatenated according to their corresponding nodes. The concatenated features are passed to the MLP for feature fusion. After obtaining the fused features for each frame, a graph is constructed by treating each frame as a node and connecting the nodes using the KNN approach. This graph is passed to the importance prediction module to further predict the frame-level importance score. The importance prediction module includes two graph convolutional layers and the MLP. The output message from the final graph convolutional layer is passed to the MLP with the sigmoid activation function to predict the importance score for each frame.

3.3.2. Keyshot selection

Following previous methods, the predicted importance score for each frame is used to obtain shot-level scores. The keyshots are selected based on shot-level scores to construct the video summary. The video is divided into non-intersecting temporal shots using the commonly used Kernel Temporal Segmentation (KTS) strategy [50]. To convert the frame-level scores to shot-level scores s_u, we take the average of frame-level scores for each shot, denoted as:

$$s_u = \frac{1}{T_u}\sum_{t=1}^{T_u} f_{u,t}, \tag{3}$$

where T_u is the number of frames in the u^{th} shot and $f_{u,t}$ is the score of the t^{th} frame in the u^{th} shot. Once the shot-level scores are obtained, the video summary is generated by selecting the most representative shots. We follow previous methods limiting the summary length to less than 15% of the original video length L. This step can be formulated as:

$$\max \sum_{u=1}^{U} b_u s_u, \quad s.t. \sum_{u=1}^{U} b_u T_u \leq 15\% \times L, \tag{4}$$

This is a typical 0/1 knapsack problem. To solve this problem, we used a dynamic programming approach. The shots with $b_u = 1$ are included in the video summary.

3.4. Learning strategy

The videos are resized to 256×256 and sampled at a rate of 2 frames per second. The KNN graphs are constructed with $K = 30$. The ReLU is used as the activation function. The training process employs the Adam optimizer with a learning rate set to 10^{-5} and a weight decay of 10^{-5}. The Mean Square Error is used as the loss function:

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2, \tag{5}$$

where $\hat{y}_n$ is the predicted score of the n^{th} frame, y_n is the ground truth score of the n^{th} frame, and N is the total number of frames in a video.

We run 5 trials, calculating the average performance across them. For each trial, the dataset is randomly split into 80% for training and 20% for testing according to DSNet [23], and the model is trained for 200 epochs.

Table 1
Performance comparisons of the proposed method with the state-of-the-art methods on F-score (%).

Method	Feature	SumMe	TVSum
SUM-GDA [12]	GoogLeNet	52.8	58.9
STVT [13]	ResNet-18	55.1	67.1
MPFN [14]	DPT ViT	51.9	62.4
CSNet [19]	GoogLeNet	51.3	58.8
STVS [20]	M-ResGoogleNet	53.6	61.9
VASNet [21]	GoogLeNet	49.7	61.4
PGL-SUM [22]	GoogLeNet	55.6	61.0
DSNet:Anchor-based [23]	GoogLeNet	50.2	62.1
DSNet:Anchor-free [23]	GoogLeNet	51.2	61.9
MHA [24]	GoogLeNet	51.1	61.0
MHA_{two} [24]	GoogLeNet+Optical flows	51.4	61.5
CAAN [25]	GoogLeNet	50.8	59.6
DAN [26]	GoogLeNet	51.8	61.7
HMT [27]	Visual+Audio	44.1	60.1
VJMHT [28]	GoogLeNet	50.6	60.9
PRLVS [29]	GoogLeNet	46.3	63.0
Argaw et al. [30]	Visual+Text	**56.9**	66.0
ADUVS [31]	Visual+Audio+Text+Aesthetics	50.4	60.7
Causalainer [32]	Visual+Text	52.4	67.5
SumGraph [35]	GoogLeNet	51.4	63.9
RSGN [36]	GoogLeNet	45.0	60.1
RCL [37]	GoogLeNet	54.0	62.6
RR-STG [38]	ResNet-50-FPN	53.4	63.0
SA-CFT [39]	GoogLeNet	56.0	62.7
GL-STGCN [40]	GoogLeNet	52.9	60.2
DSNet-MV:Anchor-based [47]	MV	51.1	59.5
DSNet-MV:Anchor-free [47]	MV	44.4	60.0
DSNet-MV:Anchor-based:no-mask [47]	MV	47.4	59.3
DSNet-MV:Anchor-free:no-mask [47]	MV	48.7	59.6
GCNCVS	MPT, MV, Res, DC, QP	53.5	**72.3**

4. Experimental results

4.1. Experimental settings

All experiments are conducted within our experimental environment, utilizing AMD Ryzen 9 5950X CPU and NVIDIA GeForce RTX 3090 GPU. An x264 encoder following the H.264/AVC video coding standard is used to obtain compressed-domain features. The proposed method is implemented using PyTorch for training and testing. Our code is available at https://github.com/lien1119/GCNCVS.

We evaluate the performance on two well-known datasets: SumMe [51] and TVSum [52]. SumMe offers 25 videos of varying content, length, and resolution across sports, holidays, and events. Videos are approximately 1 to 6 minutes in length. The ground truth summaries per video were created by 15 to 18 people. TVSum consists of 50 videos covering 10 topics including making sandwich, dog show, bee-keeping, parade, grooming an animal, flash mob gathering, attempting bike tricks, changing vehicle tire, parkour, and getting vehicle unstuck, with video durations ranging from approximately 2 to 10 minutes. The ground truth summaries per video were annotated via Amazon Mechanical Turk by 20 human annotators. Both datasets exhibit a high degree of consistency among videos. Therefore, they are suitable for training and evaluation of the proposed method.

We evaluate the performance of our proposed method and compare our method with the state-of-the-art methods on the SumMe and the TVSum datasets. The F-score, Kendall's τ correlation coefficient, Spearman's ρ correlation coefficient, and computation time serve as our evaluation metric. We compute the precision P and recall R as:

$$P = \frac{Y \cap G}{Y}, \tag{6}$$

Table 2
Performance comparisons of the proposed method with the state-of-the-art methods on Kendall's τ and Spearman's ρ correlation on the TVSum dataset.

Method	Kendall's τ	Spearman's ρ
STVT [13]	0.100	0.131
CAAN [25]	0.038	0.050
DAN [26]	0.071	0.099
HMT [27]	0.096	0.107
VJMHT [28]	0.097	0.105
PRLVS [29]	0.080	0.131
Argaw et al. [30]	0.155	0.186
ADUVS [31]	0.068	0.072
SumGraph [35]	0.094	0.138
RSGN [36]	0.083	0.090
GCNCVS	**0.157**	**0.205**

Table 3
Comparisons of average computational time (in seconds) of the proposed method with VASNet [21] and PGL-SUM [22].

Dataset	Method	Feature extraction	Frame-level importance score prediction	Total time
SumMe	VASNet [21]	4.782	0.004	4.786
	PGL-SUM [22]	4.782	0.022	4.804
	GCNCVS	-	0.416	0.416
TVSum	VASNet [21]	2.931	0.008	2.939
	PGL-SUM [22]	2.931	0.034	2.965
	GCNCVS	-	0.677	0.677

$$R = \frac{Y \cap G}{G}, \tag{7}$$

where Y is the predicted summary and G is the ground truth summary. We then compute $F-score$ as:

$$F-score = \frac{2 \times P \times R}{P+R} \times 100\% \tag{8}$$

The predicted summary is compared with ground truth summaries for each video. The model with the highest F-score on the validation split of the dataset is selected as the final well-trained model for each trial. The average F-score of trials is the final evaluation metric.

4.2. Results

We compare our method with SUM-GDA [12], STVT [13], MPFN [14], CSNet [19], STVS [20], VASNet [21], PGL-SUM [22], DSNet [23], MHA [24], CAAN [25], DAN [26], HMT [27], VJMHT [28], PRLVS [29], Argaw et al. [30], ADUVS [31], Causalainer [32], SUMGraph [35], RSGN [36], RCL [37], RR-STG [38], SA-CFT [39], GL-STGCN [40], and DSNet-MV [47]. The results are shown in Table 1. The F-score of our proposed method reaches 53.5% and 72.3% on the SumMe and the TVSum datasets, respectively. Our proposed method significantly outperforms other methods on the TVSum dataset and achieves competitive performance on the SumMe dataset compared to the best-performing method.

Moreover, we follow the rank order correlation measures in [53] using Kendall's τ and Spearman's ρ correlation coefficients to evaluate the similarity between the predicted importance scores by our proposed method and the human annotated reference scores on the TVSum dataset. The ranking of video frames is established according to the predicted frame importance scores and the human-annotated reference scores. The correlation score is obtained by averaging the individual results that compare the generated ranking to each reference ranking. We assess the performance of our method by comparing it with STVT [13], CAAN [25], DAN [26], HMT [27], VJMHT [28], PRLVS [29], Argaw et al. [30], ADUVS [31], SumGraph [35], and RSGN [36] as presented in Table 2. Our method demonstrates a high correlation with human-annotated video summaries, showcasing its effectiveness.

In addition to F-score and correlation coefficients, computation time is also a critical aspect to be considered in video summarization. Our model has approximately 21 million learnable parameters. During the training process, the time required per epoch is 13.53 seconds for the SumMe dataset and 46.56 seconds for the TVSum dataset. Note that each epoch uses 80% of the videos in the datasets as training, that is, 20 videos in the SumMe dataset and 40 videos in the TVSum dataset. We compare the average processing time for inference across all videos in the dataset required by different methods, including VASNet [21] and PGL-SUM [22], both of which use GoogLeNet as a pre-trained model for feature extraction. The results are shown in Table 3. Our proposed method takes only 0.416 seconds and 0.677 seconds on the SumMe and the TVSum datasets, respectively. The results demonstrate the effectiveness of our method in terms of efficiency and speed.

Table 4
Ablation study for each feature on F-score (%).

Feature	SumMe	TVSum
MPT	46.7	71.6
MV	**57.0**	65.5
Res	52.7	67.7
DC	45.7	69.2
QP	53.1	71.8
MPT, MV, Res, DC, QP	53.5	**72.3**

Table 5
Ablation study of the different modules on F-score (%).

Information Enhancement Module	Importance Prediction Module	SumMe	TVSum
✓	×	40.3	68.2
×	✓	51.9	70.3
✓	✓	**53.5**	**72.3**

4.3. Ablation study

To investigate the impact of each feature, we conducted the experiments on the SumMe and the TVSum datasets, as presented in Table 4. For the SumMe dataset, MV provides the highest F-score of 57.0%, while for the TVSum dataset, QP achieves the highest F-score of 71.8%. These results highlight that the contribution of the different features varies between the two datasets. Although MV performs best on the SumMe dataset, its contribution is less on the TVSum dataset. Therefore, incorporating multiple features to consider many aspects is crucial to achieve generalization across diverse datasets.

Table 5 presents the ablation study of different modules in the proposed method. When only the information enhancement module is included, the F-score is 40.3% for the SumMe dataset and 68.2% for the TVSum dataset. Conversely, when only the importance prediction module is used, the F-score shows 51.9% for the SumMe dataset and 70.3% for the TVSum dataset. The combination of both modules led to the highest F-scores, with 53.5% for the SumMe dataset and 72.3% for the TVSum dataset. This demonstrates that the integrated use of both modules provides significantly enhanced performance.

4.4. Qualitative analysis

We qualitatively evaluate our proposed method by comparing the video summarization generated by our method with ground truth, as shown in Fig. 3. The "Air Force One" video from the SumMe dataset shows the airplane landing. The 35th video from the TVSum dataset is a

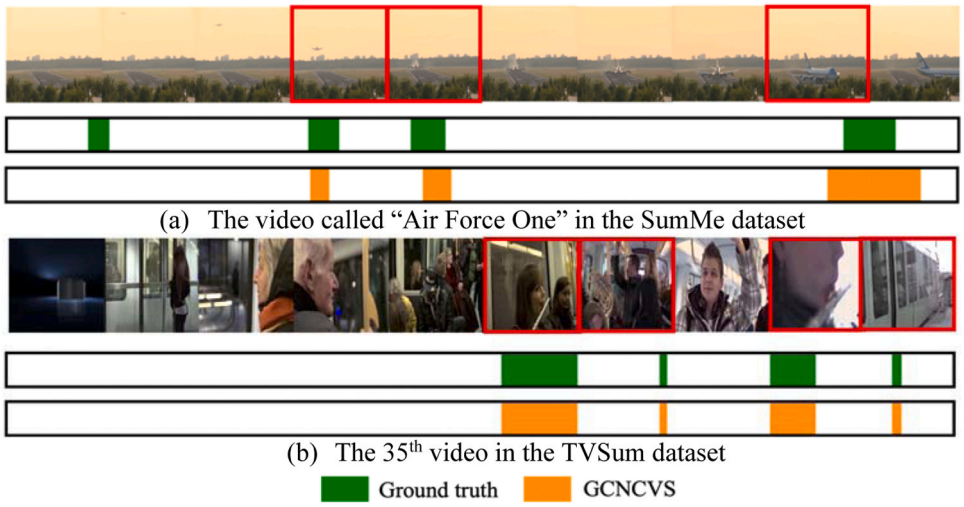

Fig. 3. Comparison of the proposed method with ground truth through qualitative evaluation. Redboxes indicate that frames are selected in the summary by the proposed method.

musical flash mob on the subway. For the "Air Force One" video, our method captures key moments such as the descent, touchdown, and turning. For the 35th video in the TVSum dataset, our method provides a ground-truth-aligned summary. Our method accurately predicts and generates summaries that closely resemble the ground truth. The comparison demonstrates the effectiveness and robustness of our method in generating high-quality and representative video summaries.

4.5. Discussion on challenges

Despite the promising performance of GCNCVS in video summarization, several challenges remain to be addressed in future work.

GCNs face challenges such as scalability and lack of interpretability. Scalability becomes an issue when summarizing long videos, which can be several hours long. In such cases, representing each frame as a node creates large graphs with numerous nodes and edges, causing increased computational complexity and memory usage. In addition, GCNs lack interpretability, making it difficult to understand how the model selects keyframes or segments in video summarization. The learning process of node embeddings and feature aggregation is hard to explain. Therefore, scalability and interpretability are urgent problems that demand further exploration.

Effective change point detection is another important issue that can greatly affect the quality of video summarization. Therefore, we would like to take advantage of the compressed domain to design a more efficient change point detection method instead of using the KTS algorithm, which is widely used in other methods.

5. Conclusion

This paper proposes GCNCVS, a novel video summarization framework that integrates various compressed-domain information as input to the graph convolutional network. The graph convolutional network effectively addresses the non-Euclidean structure of compressed-domain data and captures temporal relationships to produce an accurate and informative summary. Experimental results demonstrate the promising performance of GCNCVS, achieving an average F-score of 72.3% and surpassing other methods on the TVSum dataset. In addition, it shows Kendall's τ correlation coefficient of 0.157 and Spearman's ρ correlation coefficient of 0.205 on the TVSum dataset. The dramatically reduced processing time makes this method a promising solution for efficient and fast video summarization in video streaming environments. Additionally, the lower computational requirement of our method enables its integration into resource-constrained environments, broadening its applicability. In the future, we plan to enhance the scalability and interpretability of graph convolutional networks for video summarization tasks and leverage the compressed-domain information to design efficient change point detection.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Advanced Streaming Technology Division, Platform and Growth Center, KKCompany, Taiwan, under Grant 111K0116.

References

[1] K. Muhammad, T. Hussain, S.W. Baik, Efficient CNN based summarization of surveillance videos for resource-constrained devices, Pattern Recognit. Lett. 130 (2020), 370–275.
[2] W. Gavião, J. Scharcanski, J.M. Frahm, M. Pollefeys, Hysteroscopy video summarization and browsing by estimating the physician's attention on video segments, Med. Image Anal. 16 (1) (2012) 160–176.
[3] A. Chandrasekar, T. Radhika, Q. Zhu, Further results on input-to-state stability of stochastic Cohen-Grossberg BAM neural networks with probabilistic time-varying delay, Neural Process. Lett. 54 (2022) 613–635.
[4] T. Radhika, A. Chandrasekar, V. Vijayakumar, Q. Zhu, Analysis of Markovian jump stochastic Cohen-Grossberg BAM neural networks with time delays for exponential input-to-state stability, Neural Process. Lett. 55 (2023) 11055–11072.
[5] Y. Cao, A. Chandrasekar, T. Radhika, V. Vijayakumar, Input-to-state stability of stochastic Markovian jump genetic regulatory networks, Math. Comput. Simul. 222 (2024) 174–187.
[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1– 9.
[7] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.
[8] B. Bross, J. Chen, J.R. Ohm, G.J. Sullivan, Y.K. Wang, Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC), Proc. IEEE 109 (9) (2021) 1463–1493.
[9] K. Zhou, Y. Qiao, T. Xiang, Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward, in Proc. AAAI Conf. Artif. Intell., 2018, pp. 7582–7589.
[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Syst., 2017.
[11] B. Zhao, X. Li, X. Lu, THH-RNN: Tensor-train hierarchical recurrent neural network for video summarization, IEEE Trans. Ind. Electron. 68 (4) (2021) 3629–3637.

[12] P. Li, Q. Ye, L. Zhang, L. Yuan, X. Xu, L. Shao, Exploring global diverse attention via pairwise temporal relation for video summarization, Pattern Recognit. 111 (2021) 107677.
[13] T.C. Hsu, Y.S. Liao, C.R. Huang, Video summarization with spatiotemporal vision transformer, IEEE Trans. Image Process. 32 (2023) 3013–3026.
[14] H. Khan, T. Hussain, S.U. Khan, Z.A. Khan, S.W. Baik, Deep multi-scale pyramidal features network for supervised video summarization, Expert Syst. Appl. 237 (2024) 121288.
[15] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in Proc. Int. Conf. Learn. Represent., 2017.
[16] E. Apostolidis, E. Adamantidou, A.I. Metsai, V. Mezaris, I. Patras, Video summarization using deep neural networks: A survey, Proc. IEEE 109 (11) (2021) 1838–1863.
[17] K. Zhang, W.L. Chao, F. Sha, K. Grauman, Video summarization with long short-term memory, in Proc. Eur. Conf. Comput. Vis., 2016, pp. 766–782.
[18] B. Mahasseni, M. Lam, S. Todorovic, Unsupervised video summarization with adversarial LSTM networks, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2982–2991.
[19] Y. Jung, D. Cho, D. Kim, S. Woo, I.S. Kweon, Discriminative feature learning for unsupervised video summarization, in Proc. AAAI Conf. Artif. Intell., 2019, pp. 8537–8544.
[20] S. Kashid, L.K. Awasthi, K. Berwal, P. Saini, STVS: Spatio-temporal feature fusion for video summarization, IEEE Multimed. 31 (3) (2024) 88–97.
[21] J. Fajtl, H.S. Sokeh, V. Argyriou, D. Monekosso, P. Remagnino, Summarizing videos with attention, in Proc. Asian Conf. Comput. Vis., 2018, pp. 39–54.
[22] E. Apostolidis, G. Balaouras, V. Mezaris, I. Patras, Combining global and local attention with positional encoding for video summarization, in Proc. IEEE Int. Symp. Multimedia, 2021, pp. 226–234.
[23] W. Zhu, J. Lu, J. Li, J. Zhou, DSNet: A flexible detect-to-summarize network for video summarization, IEEE Trans. Image Process. 30 (2021) 948–962.
[24] W. Zhu, J. Lu, Y. Han, J. Zhou, Learning multiscale hierarchical attention for video summarization, Pattern Recognit. 122 (2022) 108312.
[25] G. Liang, Y. Lv, S. Li, S. Zhang, Y. Zhang, Video summarization with a convolutional attentive adversarial network, Pattern Recognit. 131 (2022) 108840.
[26] G. Liang, Y. Lv, S. Li, X. Wang, Y. Zhang, Video summarization with a dual-path attentive network, Neurocomputing 467 (2022) 1–9.
[27] B. Zhao, M. Gong, X. Li, Hierarchical multimodal transformer to summarize videos, Neurocomputing 468 (2022) 360–369.
[28] H. Li, Q. Ke, M. Gong, R. Zhang, Video joint modelling based on hierarchical transformer for co-summarization, IEEE Trans. Pattern Anal. Mach. Intell. 45 (3) (2023) 3904–3917.
[29] G. Wang, X. Wu, J. Yan, Progressive reinforcement learning for video summarization, Inf. Sci. 655 (2024) 119888.
[30] D.M. Argaw, S. Yoon, F.C. Heilbron, H. Deilamsalehy, T. Bui, Z. Wang, F. Dernoncourt, J.S. Chung, Scaling up video summarization pretraining with large language models, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2024, pp. 8332–8341.
[31] H. Huang, Z. Wu, G. Pang, J. Xie, An aesthetic-driven approach to unsupervised video summarization, IEEE Access 12 (2024) 128768–128777.
[32] J.H. Huang, C.H.H. Yang, P.Y. Chen, M.H. Chen, M. Worring, Causalainer: causal explainer for automatic video summarization, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2023, 2630–2636.
[33] J. Wu, S.H. Zhong, Y. Liu, MvsGCN: A novel graph convolutional network for multi-video summarization, in Proc. ACM Int. Conf. Multimedia, 2019, pp. 827-835.
[34] J. Wu, S.H. Zhong, Y. Liu, Dynamic graph convolutional network for multi-video summarization, Pattern Recognit. 107 (2020) 107382.
[35] J. Park, J. Lee, I.J. Kim, K. Sohn, Sumgraph: Video summarization via recursive graph modeling, in Proc. Eur. Conf. Comput. Vis., 2020, pp. 647–663.
[36] B. Zhao, H. Li, X. Lu, X. Li, Reconstructive sequence-graph network for video summarization, IEEE Trans. Pattern Anal. Mach. Intell. 44 (5) (2022) 2793–2801.
[37] Y. Zhang, Y. Liu, P. Zhu, W. Kang, Joint reinforcement and contrastive learning for unsupervised video summarization, IEEE Signal Process. Lett. 29 (2022) 2587–2591.
[38] W. Zhu, Y. Han, J. Lu, J. Zhou, Relational reasoning over spatial-temporal graphs for video summarization, IEEE Trans. Image Process. 31 (2022) 3017–3031.
[39] R. Zhong, R. Wang, W. Yao, M. Hu, S. Dong, A. Munteanu, Semantic representation and attention alignment for graph information bottleneck in video summarization, IEEE Trans. Image Process. 32 (2023) 4170–4184.
[40] G. Wu, S. Song, J. Zhang, Global-local spatio-temporal graph convolutional networks for video summarization, Comput. Electr. Eng. 118 (2024) 109445.
[41] M. Basavarajajiah, P. Sharma, Survey of compressed domain video summarization techniques, ACM Comput. Surv. 52 (6) (2019) 1–29.
[42] C.M. Chew, M.S. Kankanhalli, Compressed domain summarization of digital video, in Proc. Adv. Multimedia Inf. Process., 2001, pp. 490–497.
[43] J.C.S. Yu, M.S. Kankanhalli, P. Mulhen, Semantic video summarization in compressed domain MPEG video, in Proc. Int. Conf. Multimedia Expo., 2003, pp. 329–332.
[44] J. Ren, J. Jiang, Y. Feng, Activity-driven content adaption for effective video summarization, J. Vis. Commun. Image Represent. 21 (8) (2010) 930–938.
[45] F. Wang, C.W. Ngo, Summarizing rushes videos by motion, object, and event understanding, IEEE Trans. Multimed. 14 (1) (2012) 76–87.
[46] P. Dong, Y. Xia, D.D. Feng, Real-time storyboard generation for H.264/AVC compressed videos, in Proc. IEEE Int. Conf. Multimedia Expo., 2012, pp. 544–549.
[47] Lakshya, S. Agarwal, V.S. Kota, M.R. Voleti, DSNet-MV: Fast summarization of surveillance video's using deep learning in compressed domain using motion vectors, in Proc. IEEE India Counc. Int. Conf., 2021.
[48] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, A. Luthra, Overview of H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol. 13 (7) (2003) 560–576.
[49] G.L. Miller, S.H. Teng, W. Thurstion, S.A. Vavasis, Separators for sphere-packings and nearest neighbor graphs, J. ACM 44 (1) (1997) 1–29.
[50] D. Potapov, M. Douze, Z. Harchaoui, C. Schmid, Category-specific video summarization, in Proc. Eur. Conf. Comput. Vis., 2014, pp. 540–555.
[51] M. Gygli, H. Grabner, H. Riemenschneider, L.V. Gool, Creating summaries from user videos, in Proc. Eur. Conf. Comput. Vis., 2014, pp. 505–520.
[52] Y. Song, J. Vallmitjana, A. Stent, A. Jaimes, TVSum: Summarizing web videos using titles, in Proc. IEEE Conf. Compt. Vis. Pattern Recognit., 2015, pp. 5179–5187.
[53] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, Rethinking the evaluation of video summaries, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7596–7604.

Chia-Hung Yeh is currently a Distinguished Professor at National Taiwan Normal University, Taipei, Taiwan. He has coauthored more than 300 technical international conferences and journal papers and held 47 patents in the USA, Taiwan, and China. His research interests include deep learning, video coding, 3D reconstruction, image/video processing. Dr. Yeh was the Associate Editor for the Journal of Visual Communication and Image Representation, the EURASIP Journal on Advances in Signal Processing, the International Journal of Pattern Recognition and Artificial Intelligence, and the ICT Express. He was the recipient of the 2013 IEEE MMSP Top 10% Paper Award, the 2014 IEEE GCCE Outstanding Poster Award, the 2015 APSIPA Distinguished Lecturer, the 2017 IEEE SPS Tainan Section Chair, the 2017 Distinguished Professor Award of NTNU, the IEEE Outstanding Technical Achievement Award (IEEE Tainan Section). He became a Fellow of IET in 2017.

Chih-Ming Lien received his B.S. degree from the Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan, in 2022. He is pursuing the Ph.D. degree in Department of Electrical Engineering from National Taiwan Normal University, Taipei, Taiwan. His research interests include video coding and deep learning for video processing.

Zhi-Xiang Zhan received the B.S. degree from the Department of Electrical Engineering in National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan in 2021 and the master's degree in National Sun Yat-sen University, Kaohsiung, Taiwan in 2023. His research focuses on deep learning for image/video processing and computer vision.

Feng-Hsu Tsai graduated with a master's degree at Computer Science and Information Engineering in National Taiwan University. He is working as a senior manager in Advanced Streaming Technology Division, Platform and Growth Center, KKCompany, to lead the advanced technology R&D team on per-title encoding, perceptual streaming engine, low latency live, and ML.

Mei-Juan Chen received her B.S., M.S. and Ph.D. degrees in Electrical Engineering from National Taiwan University, Taipei, in 1991, 1993, and 1997, respectively. She is a professor at the Department of Electrical Engineering, National Dong Hwa University, Taiwan. Her research interests include image/video processing and compression.