FRC-GIF: Frame Ranking-based Personalized Artistic Media Generation Method for Resource Constrained Devices

Ghulam Mujtaba† *Senior Member, IEEE*, Sunder Ali Khowaja† *Senior Member, IEEE*, Muhammad Aslam Jarwar *Senior Member, IEEE*, Jaehyuk Choi, and Eun-Seok Ryu, *Senior Member, IEEE*

***Abstract*—Generating video highlights in the form of animated graphics interchange formats (GIFs) has significantly simplified the process of video browsing. Animated GIFs have paved the way for applications concerning streaming platforms and emerging technologies. Existing studies have led to large computational complexity without considering user personalization. This paper proposes lightweight method to attract users and increase views of videos through personalized artistic media, i.e., static thumbnails and animated GIF generation. The proposed method analyzes lightweight thumbnail containers (LTC) using the computational resources of the client device to recognize personalized events from feature-length sports videos. Next, the thumbnails are then ranked through the frame rank pooling method for their selection. Subsequently, the proposed method processes small video segments rather than considering the whole video for generating artistic media. This makes our approach more computationally efficient compared to existing methods that use the entire video data; thus, the proposed method complies with sustainable development goals. Furthermore, the proposed method retrieves and uses thumbnail containers and video segments, which reduces the required transmission bandwidth as well as the amount of locally stored data. Experiments reveal that the computational complexity of our method is 3.73 times lower than that of the state-of-the-art method.**

I. Introduction

Video sharing has increased over the past decade. Statistics concerning upload volume on YouTube platforms show the indescribable interest of end users towards video modalities [1]. However, this diversity and variety make it difficult for creators to compel users to watch the content, let alone click for fast viewing. The first impression of any video content is represented by its thumbnail and title, which guarantee at least one view from the user. Streaming sites and video publishers consider the use of creative and attractive image thumbnails as a major factor for increased click-through rate (CTR). In this regard, several researchers strive to automatically generate thumbnail images from videos to boost the CTR [2]–[4].

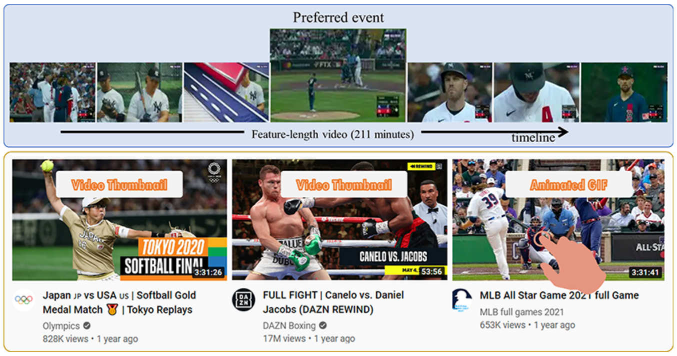

Fig. 1. Artistic media in the form of static thumbnails and animated GIF images is used in popular streaming platforms to highlight recommended videos. Generally, the most preferred events are selected as artistic thumbnails according to the video category to attract users to get more views (above). Animated GIFs are played whenever a user hovers over the static thumbnail (below).

Animated graphics interchange formats (GIFs) are an evolution of an image thumbnail that unveils selected frames from a video in a continuous loop without any sound. With the increasing popularity of social media networks, especially Reddit and Tumblr, automated GIFs have gained popularity, particularly when creating Cinema graphs or Internet memes [5]. Several online platforms, including Ezgif, Imgflip, and GIFSoup, provide easy-to-use tools for converting videos into GIF formats. Nevertheless, the generation of a GIF remains a manual process that requires a user to specify the timestamps of the beginning and end of the video clip. The manual input of exact time ranges and human intervention makes it difficult to adopt existing solutions using large-scale video-processing platforms such as YouTube.

The demand for generating animated GIFs in an automated manner has been instigated by streaming platforms, which aim to deliver a succinct preview of video content within a time span of 3 to 15 seconds [5]. As shown in Figure I, this encompasses the presentation of artistic media pertaining to sports videos: (1) an animated GIF when the user hovers the mouse on a static thumbnail (below) and (2) the selection of prominent frames from the feature-length video (above). It is noteworthy that the animated GIF serves as the primary gateway through which users make informed decisions about whether to watch a feature-length video. As a result, there has been a significant increase in interest in the area of automated

G. Mujtaba is with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, USA. E-mail: gmujtabakorai@gmail.com

S.-A Khowaja is with the Department of Telecommunication Engineering, University of Sindh, Pakistan. E-mail: sandar.ali@usindh.edu.pk

M.A Jarwar is with the Department of Computing, Sheffield Hallam University, Sheffield, United Kingdom. E-mail: a.jarwar@shu.ac.uk

J. Choi is with the Department of Software, Gachon University, Republic of Korea. E-mail: jchoi@gachon.ac.kr

E.-S. Ryu is an Associate Professor at the Department of Immersive Media Engineering at Sungkyunkwan University (SKKU), Republic of Korea. E-mail: esryu@skku.edu

GIF creation from both academic and industrial sectors.

The generation of automated GIF corresponds to physical computer vision problems such as video summarization [6], video highlights [7], [8], creativity [9], and visual interestingness [10], [11], but differs in the sense that distinct spatio-temporal visual patterns and various forms of emotions are needed to generate a series of images that will loop forever. Furthermore, unlike the aforementioned computer vision fields, automated GIFs are in real-world demand and are associated with applications such as video promotion, video sharing, advertising, and photojournalism. For such real-world applications, we must consider user preferences to improve a video's CTR [12]. Therefore, the incorporation of user preferences in the context of automated GIF generation remains a research challenge and an open research problem.

‘ **Computational Constraints**: A standard video is comprised of a continuous vast collection of frames capturing various aspects such as character appearances, movements, object interactions, events, and scenes [13]. This extensive volume of frames inherently imposes considerable computational demands during processing, especially for feature-length videos. However, current methods of processing the entire video/frames exhibit inefficiencies primarily due to frame redundancy within short time intervals (often only seconds) [2], [14]. Moreover, it is imperative to reduce the resolution of high-definition video, which further adds a layer of computational complexity. This compounded computational burden renders such approaches less viable for resource-constrained devices, extending processing duration significantly. When it comes to creating personalized artistic media through server-based methods, it's important to keep in mind that there may be limitations due to computing power. While advancements in computational power and data parallelism techniques have expanded the possibilities, large user bases may still pose a challenge for simultaneous generation [15]. Therefore, there is an urgent need to develop a lightweight artistic media generation method that addresses these challenges and is meticulously designed to cater to the intricacies of feature-length videos.

Privacy Constraints: Creating personalized artistic media is a difficult task because it is subjective and depends on individual preferences. In response to this challenge, server-side technologies have emerged as a promising avenue for addressing these issues [16]. While there are advantages to using server-side technology, adoption raises some significant considerations. For example, artistic media generation methods require access to user data, and using server-based technology raises data privacy concerns. Additionally, safeguarding consumer privacy is more crucial, especially when it comes to server-side components that handle data retrieval and recommendations. Client-based techniques can address these issues associated with server-side technologies. The computational complexity can be reduced during automated artistic media generation by adopting lightweight container-based methods [17], [18]. These methods also prioritize personalization, which aligns with sustainable development goals and creates a human-centered solution.

Personalization Effect: Existing studies either use statistical measures to achieve the personalization effect or completely ignore the given aspect for automated GIF generation [2]. The study [17], [18] introduced the concept of using action recognition to achieve personalization. The selected thumbnails will undergo an action classification module in order to select the thumbnails, which represents intended action. In this regard, the proposed method dwells on the idea to integrate action recognition module for improving the user experience through personalization.

In order to address the task of automated GIF generation while considering the personalization and minimizing computational complexity, we proposed a lightweight container (LTC)-based client-driven automated GIF generation method. This method focuses specifically on generating GIFs from sports videos and was first introduced in the conference paper version [19]. The proposed approach extends the work on both fronts to generate GIFs in a faster and more efficient manner. For example, we utilized LTC and frame ranking methods to detect key events in sports videos, such as penalty shots in soccer, to reduce the overall processing time. We introduce the use of the ConvNeXt base model [20] integrated with a triplet attention module [21] to learn cross-dimensional features and improve the event recognition performance. Although vertex pooling intuitively maps the memory pool of interleaved vertex data, which should be effective in generating automated GIFs, we show that the proposed network intrinsically learns multi-scale features, thereby performing better and requiring a lower computational time. The proposed method was rigorously validated using a dataset comprising 23 publicly broadcasted sports videos, thereby substantiating its efficacy and overall effectiveness. The main contributions of this study are succinctly summarized as follows:

- We propose a novel lightweight client-driven frame-ranking method for the automated generation of personalized GIFs extracted from feature-length sports videos.
- We investigate 23 feature-length sports videos, totaling approximately 2,818.96 minutes in duration, spanning six distinct sports categories, namely baseball, basketball, boxing, cricket, football, and tennis.
- We propose ConvNeXt and a triplet attention module-based convolutional network for LTC analysis, with the primary objective of classifying personalized events.
- We achieved state-of-the-art results, both in terms of computational efficiency and accuracy, i.e., 3.69 times faster and $> 10\%$ accuracy, respectively.

To the best of our knowledge, this is the first attempt at generating personalized GIFs using LTC and frame-ranking methods on end-user devices for streaming platforms [1].

The subsequent section of this paper is organized as follows: Section II offers an extensive review of the existing literature. Section III delineates the proposed client-driven and frame-ranking-based method for the selection of thumbnails, along with the details of the ConvNeXt and Triplet attention module network. Section IV discusses the qualitative and quantitative results of the feature-length sports videos. Moreover, it en-

[1] The code and trained models are publicly available on GitHub at https://github.com/iamgmujtaba/FRC-GIF.

compasses the execution of various ablation studies and comparative analyses to substantiate the efficacy of the proposed approach. Finally, concluding remarks are presented in Section V.

II. Related Works

This section provides a brief review of the existing methods that focus on video understanding, artistic thumbnail generation, animated GIF generation, and resource-constrained methods.

A. Video Understanding Methods

Video understanding is a prominent field in computer vision research. Methods for video understanding have been extensively explored through the application of temporal action localization [22] and human action recognition [6]. The latter involves recognizing events from a cropped video clip, which is accomplished through various methods such as multistream networks [23], recurrent neural networks [24], and a combination of convolutional and recurrent neural networks [6]. [25] used a two-stream structure and extended it to a 3D CNN for application in human action recognition. This was obtained by pretraining a 2D CNN model using the ImageNet dataset [26] and extending the 2D CNN model to a 3D CNN by repeated weighting in a depth-wise manner. The features extracted using the bag-of-words method and pretrained networks were then used as local descriptors to train the 3D CNN network structure. Most existing methods use temporal segments [27] to prune and classify videos. Recent studies have shifted their focus to exploiting contextual information to improve event recognition further. Context represents and utilizes both spatio-temporal information and attention, which helps in learning adaptive confidence scores to utilize the surrounding information [28]. Other methods utilize time integration and motion-aware sequence learning, such as long short-term memory (LSTM) [29]. Attention-based models have also been used to improve the integration of spatio-temporal information [30].

B. Artistic Thumbnail Generation Methods

Delivering influential thumbnails is essential for obtaining user attention. Traditionally, automatic thumbnails were generated based on a random selection of fixed frames from the corresponding video, that is, the first or median frame. Consequently, the recommended thumbnails were purposeless and did not provide relevant context. Most traditional approaches are based on preserving thumbnail characteristics from purely visual content and ignoring queries or metadata. A probabilistic sampling technique was used to obtain thumbnails that summarize prominent objects and their dynamics in [31] while identifying clusters and tracking areas of the clip that exhibit affine motion coherence of the frames. In [2], the authors proposed the HECATE method that uses subjective and objective metrics such as the visual and aesthetic qualities of a frame to select attractive thumbnails. Following the aforementioned study, several machine learning and deep learning techniques were proposed that focused on capturing high-level visual features from important objects, people, and subjects [32], [33]. Another direction that researchers have attempted to explore for thumbnail generation is the extraction of semantic information from the text associated with video metadata [34]. A multitasking deep visual semantic-embedded thumbnail generation model was proposed to measure the relationships between queries and frames by extruding them into a common latent semantic space [35]. Appropriate thumbnails and tags were generated for the corresponding video by exploiting image captioning to infer semantic information from the visual features [36]. A deep learning-based thumbnail selection method was proposed in [37] by combining adversarial and reinforcement learning methods. Thumbnail selection was based on the rating of visual content representatives and the aesthetic quality of the video frame. In [38], authors proposed a new thumbnail generation method that considers user queries and removes semantically similar content to generate thumbnails. Personalization is achieved through user inputs to understand and retrieve news. In contrast to this method, the proposed work is fully automatic and uses key-frame selection to generate thumbnails. In [14], authors proposed a thumbnail selection video using various modules such as face detection, close-up detection, and logo detection. However, the objective is not to generate a personalized automated GIF, but rather to enhance the quality of the thumbnails by eliminating blurry frames or enhancing image quality analysis. Moreover, the use of various modules increases the computational complexity of the overall system while increasing the number of hyperparameters that need to be optimized. Previous methods required enormous computational resources to analyze entire video data (frames) and identify personalized content from feature-length videos. Nevertheless, the proposed method necessitates a small amount of data (thumbnails) to achieve better results and offer instantaneous solutions.

C. Animated GIF Generation Methods

Animated GIF images were first created in 1987; however, their applications have been widely explored in recent years. Specifically, in [5], animated GIFs were reported to be more attractive than other forms of media, including photos and videos on social media platforms such as Tumblr. They identified GIF features that contribute to fascinating users, such as animations, storytelling capabilities, and emotional expressions. In addition, several studies [39], [40] devised methods to predict viewers' sentiments towards animated GIFs. Despite viewer engagement, in [41], it was concluded that viewers may have diverse interpretations of animated GIFs used in communication. They predicted facial expressions, histograms, and aesthetic features and then compared them to those in [40] to determine the most appropriate video features for expressing useful emotions in GIFs. Meanwhile, the authors of [13] proposed a personalized human-character-oriented animated GIF generation method for visual songs using a feed-forward 2D deep neural network while utilizing the computational resources of the end-user device. In another approach presented in [42], sentiment analysis was used to estimate the annotated GIF text and visual emotion scores.

By combining 3D and 2D convolutions to model temporal information, the SinGAN-GIF technique was proposed to train deep generative networks on a single GIF or short video clip [43]. From an aesthetic perspective, in [2], frames were selected by measuring various subjective and objective metrics of video frames (such as visual quality and aesthetics) to generate GIFs. In a recent study [44], the authors proposed a client-driven method to mitigate privacy issues while designing a lightweight method for streaming platforms to create GIFs. Instead of adopting full-length video content in their method, they used acoustic features to reduce the overall computational time required for resource-constrained devices. A transformer-based encoder–decoder structure was proposed for the automatic captioning of GIFs. It uses pre-trained generic feature representations of video captions and other downstream tasks such as sentence localization and question answering in the video [45]. Another new approach was proposed to automatically create GIF thumbnails of videos and improve the CTR for newly uploaded videos using the generative variational dual encoder model [16]. The authors also introduced a new dataset with 1070 videos and 5394 corresponding annotated GIFs.

D. Resource Constrained Methods

Recently, several authors proposed CNN architectures suitable for processing-constrained devices [46]. However, analyzing an entire video data (frame) to obtain personalized content from feature-length videos requires significant computational resources. Thus, resource-constrained devices cannot analyze video data in near or real-time [17]. Therefore, real-time video content detection using resource-constrained devices remains challenging. The authors in [17] proposed a novel method for generating movie trailers using a container of thumbnails for resource-constrained devices. Another method proposed by creating animated GIFs based on the gender of a human character increased the CTR of video songs on a streaming platform [13]. Recently, a personalized lightweight thumbnail container-based keyshot summarization (LTC-SUM) method was proposed to handle computation and privacy bottlenecks in resource-constrained devices [18]. In this study, we proposed a new lightweight method that can generate personalized thumbnails and animated GIFs simultaneously using resource-constrained devices.

Researchers have proposed several methods that handle thumbnail [14] and animated GIF generation processes separately [2], [44]. HECATE is a prominent approach used for generating thumbnails and animated GIFs from entire video/frame [2]. However, generating thumbnails and animated GIFs requires enormous computational resources. Modern devices used by regular users have limited processing capabilities, which makes it time-consuming to generate thumbnails and animated GIFs using entire video/frames on the client device. Additionally, the method that uses audio cues for generating animated GIFs may not accurately detect personalized actions/objects [44]. There is a lack of lightweight techniques that can generate thumbnails or animated GIFs for resource-constrained devices. Lightweight client-driven techniques for generating artistic media are still in the early stages of development, and more effective methods are required to bridge the semantic gap between video understanding and personalization. Additionally, most modern client devices have limited computational capabilities. Moreover, inspecting full-length videos to create artistic media is time-consuming and unreasonable for real-time solutions [2].

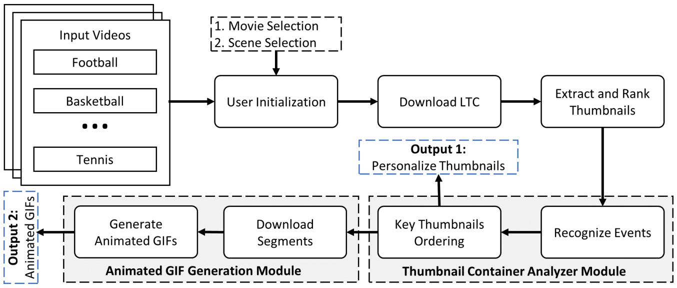

Fig. 2. The overall process of the proposed client-driven FRC-GIF generation method.

III. FRC-GIF

This study proposed FRC-GIF, a frame-ranking and container-based personalized animated GIF generation technique using artistic media. The semantic workflow of the proposed FRC-GIF method for generating artistic media is shown in Figure 2. The proposed method first uses an initialization method for artistic media selection through user initialization block. Once selected, lightweight containers (LTC) are initialized and artistic thumbnails are obtained. The HTTP Live Streaming (HLS) server is configured to obtain the LTC as proposed in the study [17]. The thumbnail containers were collected individually from the source video using FFmpeg [47] in the streaming servers. Every thumbnail container has 25 thumbnails. Most of the existing studies consider the first frame of a video for the thumbnail selection, however, in this work, we propose the rank-pooling method [48] for the frame ranking to select the subsequent thumbnail. The thumbnails are then merged into 5×5 containers according to their playtime. A single thumbnail and thumbnail container depicts the playback time of the corresponding video as 1 and 25 seconds, respectively. The entire duration of the source video is covered by sequences of thumbnail containers. The sizes of each thumbnail and thumbnail container was fixed at 160×90 and 800×450 pixels, which is in compliance with previous studies [17]. The timestamp information in the first phase *Thumbnail analyzer container module* generates an artistic thumbnail from the given video segment in the first phase of the proposed method. The second phase includes the *Animated GIF Generation* module that downloads the segments and generates user-personalized animated GIFs with an adequate processing time duration on resource-constrained client devices such as the NVIDIA Jetson TX2, respectively. Algorithm 1 shows the processing steps required to generate personalized thumbnails and animated GIFs from a video using the proposed approach. The proposed method and its components are described in the subsequent subsections.

```
Data: Input: Thumbnail containers
Result: Generated Artistic Media
Initialization:- $N$: Number of thumbnails ($T$) inside
  thumbnail containers ($LTC$)
- Personalize events ($P$); Segments ($S$); Threshold $\geq$
  80
Main Loop: while $i < N$ do
    Extract and rank $T$ from $LTC$
    $keyTList \leftarrow$ determineEvents($T$, $P$, Threshold)
    Identify the number of segments ($S$) from the list
    Download $S$
    Generate animated GIF from $S$ (Output 2)
end
Function determineEvents($T$, $P$, Threshold)
    Analyze $T$ as per $P$
    Perform key $T$ ordering (Output 1)
    Prepare key $T$ list
    return key $T$ list
```

Algorithm 1: Algorithm for analyzing events from thumbnail containers to generate personalized thumbnails and associated GIFs.

A. Ranking Thumbnails

As suggested in the preceding section, the conventional method for thumbnail selection involves the consideration of the first frame from the container or calculating a mean image. In the context of this study, we introduce a more efficient and alternative approach for selecting thumbnails from each container. The identification of keyframes, or the process of ranking individual frames, remains an ongoing and multi-faceted subject of study. For instance, in some applications, the appearance information seems to be a better fit, whereas long-term dynamics are considered a key aspect for others. In this study, we demonstrated the following: the rank pooling method proposed in [48], can be used to rank the frames to select a thumbnail. Let us denote the video frames as $\mathbf{v_1}, ..., \mathbf{v_t}$, where t represents the number of frames. The rank pooling method represents a set of frames as a ranking function. The feature vector extracted from each frame v_t is denoted by $\theta(v_t)$ and the time average for all the features extracted from individual frames up to time t is represented by $\mathbf{A_t = 1/t \sum_{t=1}^{T}} \theta(\mathbf{v_t})$. The function assigns a ranking score to each frame associated with time t, such that $\nabla(t|p) = \langle p, A_t \rangle$, where p represent the parameter vector. The rank scores for each frame were reflected by the function parameter p during the learning process. Earlier times are associated with larger scores and vice versa; that is $\mathbf{q < t} \Rightarrow \nabla(q|p) < \nabla(t|p), \forall\{q, t\}$. The problem of learning the parameter vector p can be considered a convex optimization problem that can be solved using the RankSVM method [49], as shown in Equation 1.

$$
\begin{aligned}
p^* &= \arg\max_{p} Z(p), = \phi(v_1, ..., v_t; \theta) \\
Z(p) &= \frac{\sigma}{2}\|p\|^2 + \frac{2}{t(t-1)} \\
&\quad \times \sum_{q<t} \min(1 - \nabla(q|p) + \nabla(t|p), T)
\end{aligned}
\tag{1}
$$

The former expression in Equation 1 represents the regularization that is normally used in SVMs, whereas the latter example represents the hinge loss that computes the number of incorrect rankings yielded by the scoring functions, that is, $\mathbf{q < t}$. By definition, an incorrect ranking is considered when a pair yields scores less than a unit margin $\nabla(t|p) > \nabla(q|p)+1$. The optimizer in Equation 1 provides a descriptive vector p^* via a mapping function ϕ that maps the t video frames to corresponding vector, respectively. The vector provides sufficient information to rank the frames accordingly. However, we proceed with the aggregation of the ranking information to obtain the video descriptor. Existing rank-pooling methods use features such as motion boundary histograms, histograms of optical flows, improved dense trajectories, and histograms of gradients [48]. In this study, we used RGB frames (image pixels) directly for the aforementioned function. In this work, we apply Fisher vector coding on the feature vectors obtained using improved dense trajectories, motion boundary histograms, histogram of optical flows, and histogram of oriented gradients to obtain a single descriptor vector, respectively. Therefore, $\theta(v_t)$ acts as an operator that stacks the pixel components from an RGB frame on a large vector. Subsequently, the vector p^* is a descriptor that yields the number of elements equivalent to a single RGB frame and thus can be considered a standard RGB image. It also suggests that the number of elements in p^*, $\theta(v_t)$, and v_t are the same, respectively. Furthermore, we summarize the information from the video frames of the container. We then use vector p^* and compute the gradient with each frame. The use of RGB frames for applying rank pooling to action-based videos has two benefits. The gradient in this study refers to the directional change in color intensity values, accordingly. One benefit is the decrease in processing time since optical flows require considerable computational power [6], and the second is the performance improvement compared to using optical flows, which is compliant with the observations in study [6]. In this study, the frame that yielded the least gradient response was considered the thumbnail. Similar to the RGB modality, there are two main reasons for selecting the least-gradient response. The first is the effectiveness of the least gradient response against color values for optimizing loss functions, as highlighted in [50], and the second is the sensitivity of GIF images towards image gradients owing to drastically different signatures in the form of dotted patterns, false contours, and flat regions [50]. Once the key frame, i.e. with the least gradient response, is selected, the frame is then passed as an input to the backbone network, i.e., ConvNeXt, for subsequent learning process, respectively.

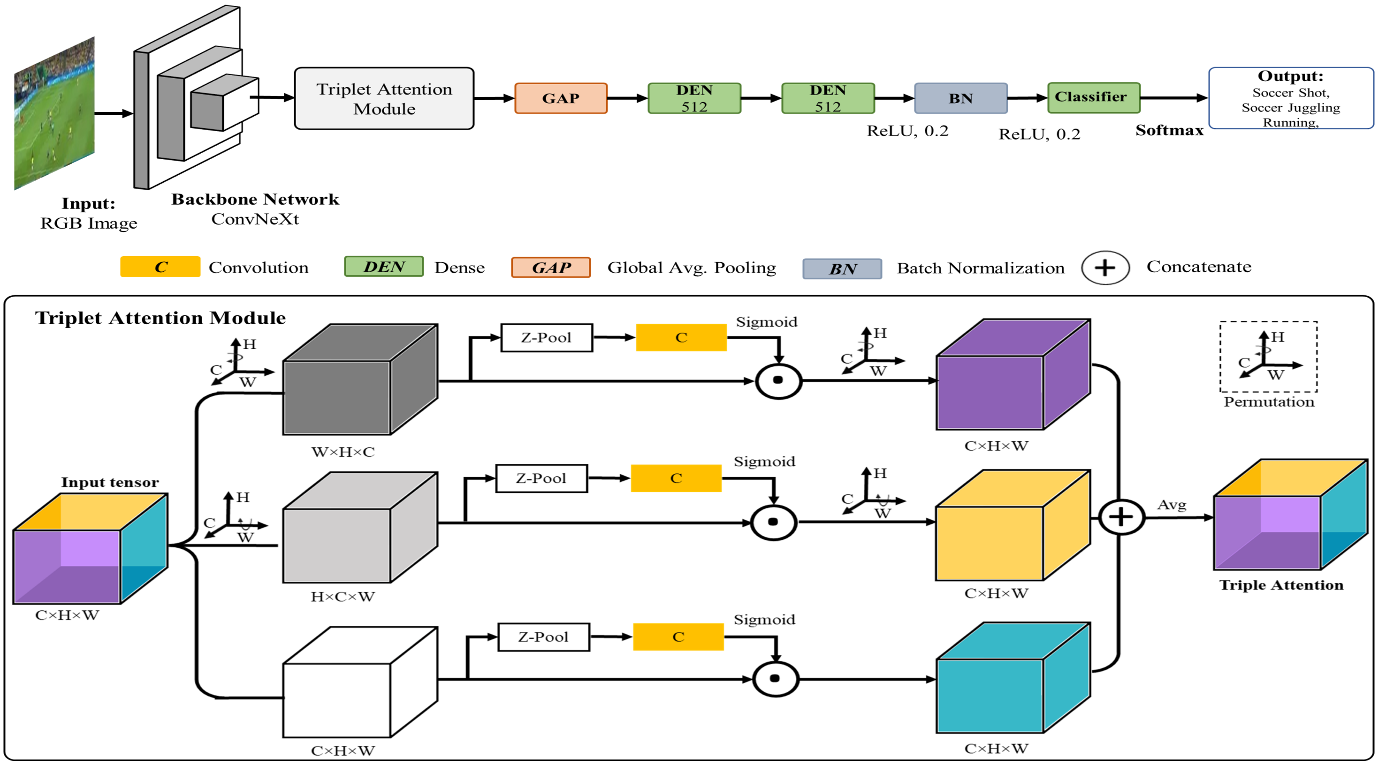

Fig. 3. The proposed architecture of the 2D convolutional neural network. It analyzes each extracted thumbnail image as the input. The most preferred detected events from thumbnail images are classified as the output.

B. Thumbnail Container Analyzer Module

The thumbnail container analyzer module consists of three segments: a backbone feature extractor, an attention module, and a classifier, as shown in Figure 3. The backbone feature extractor utilizes a pre-trained CNN model, i.e., the ConvNeXt base model [20] pre-trained on the ImageNet database [26], to extract high-level semantic features from the thumbnails. The features are then fed to an attention module to extract contextual information from the high-level semantic features. The triplet attention module (TAM) is discussed in detail in the following subsection.

The system initiated by allowing the user to select the sports video and configure their preferences by specifying preferred scenes. Subsequently, it download the LTC from the HLS server associated with the chosen video. Then, it extract and rank thumbnails from the downloaded LTC. The thumbnail container analyzer module analyzes each extracted thumbnail individually based on the event(s) and user preferences. The proposed method selects a personalized artistic thumbnail from an analyzed LTC. To ensure the quality of the generated media, we have a rigorous selection process for artistic thumbnails. We set a quality threshold, and only thumbnails that surpass a score of 80 are considered for further processing. These selected thumbnails are then earmarked for inclusion in the *Key thumbnail ordering*. It is a crucial step in our workflow. It involves carefully assembling and encoding these high-quality thumbnails within a text-based timestamp information file. The data within these artistic thumbnail files is carefully arranged and ranked in chronological order to enhance temporal consistency. The order provides viewers with a seamless and logical progression of events in the video. The personalized thumbnail is selected as the first output in the key-ordered text file. The second output that can generate animated GIFs is described in Section III-C. The algorithm 1 illustrates the processing steps required to generate thumbnails and animated GIFs from videos using the proposed method.

1) Triplet Attention Module: In this study, a TAM [21] is incorporated before the global average pooling (GAP) layer. The workflow of TAM is shown in Figure 3. The characteristics of TAM are defined by three parallel modules that capture the cross-dimensional interaction between the width and channel dimensions, the interaction between the height and channel dimensions, and the interaction between the width and height. The interactions in the last module, that is width and height, can be which is considered the spatial attention. We present a computation to explain the dynamics of the cross-dimensional attention map obtained from the interactions between the channel and width modules. Let us denote the feature map obtained from the ConvNeXt backbone network as $\Psi_i \in \mathbf{R}^{C \times H \times W}$, where C, H, and W, refer to the channel, height, and width, respectively, the feature map. To obtain the cross-dimensional interaction between the channel and width, the branch first applies max pooling and average pooling with respect to the height dimension, followed by their concatenation. The pooling process is expressed by Equation 2:

$$\Psi_{b1} = concat(pool^H_{max}(\Psi_i), pool^H_{avg}(\Psi_i)) \tag{2}$$

where Ψ_{b1} represents the operation of the TAM associated with the first branch. The expression in Equation 2 yields the pooled features Ψ_{b1} with dimensions $\mathbf{R}^{2\times C\times W}$. The resultant feature map underwent convolutional operations with a 7×7 convolution layer, followed by a sigmoid function, as shown in Equation 3.

$$\mathcal{F}_{b1} = \sigma(\rho(\Psi_{b1})) \tag{3}$$

where σ and ρ refer to the sigmoid activation and convolutional function, respectively. The same process is applied to obtain $\mathcal{F}_{b2}$ and $\mathcal{F}_{b3}$ with respect to the width and channel dimensions, respectively. The output features obtained from the TAM can be represented by Equation 4.

$$\Psi_{TAM} = \frac{(\Psi_i \odot \mathcal{F}_{b1} + \cdots + \Psi_i \odot \mathcal{F}_{bn})}{bn} \tag{4}$$

In the above equation, the notation bn refers to the number of branches in the TAM, whereas the $\odot$ notation represents the element-wise multiplication operation. TAM was employed because of its organic fusion of channels and spatial attention without losing spatial information. Furthermore, as this study focuses on lightweight personalized GIF generation, TAM is beneficial in that it is a lightweight module that outputs a single channel in comparison to its contemporaries such as channel attention [51] and convolutional block attention modules [51].

C. Animated GIF Generation Module

In our artistic animated GIFs generation module, timestamp information extraction is a crucial component initiated during the key thumbnail ordering phase. This innovative approach eliminates the need to download entire video files, optimizing resource utilization. By harnessing the timestamp data, we precisely targeted and selectively retrieved specific video segments from an HLS server. This selective downloading approach minimizes data transfer and computational resources. Once these video segments are acquired, we employ [47] to process and transform them into individual animated GIFs. It is important to emphasize that each video segment corresponds to a distinct GIF output. This segmented approach enables the efficient and resource-conscious process of creating animated GIFs for all relevant segments of the video.

IV. Results and Discussion

This section presents an extensive experimental evaluation of both the baseline and proposed approaches. Initially, we explain the experimental configuration, encompassing the utilization of a video dataset, foundational techniques, and hardware specifications. Following this, we undertake objective evaluations by analyzing the event recognition model to scrutiny via well-known approaches. Subsequently, we examine the performance of the proposed method from various perspectives, drawing comparisons against the baseline methods. Then quantitative comparisons between the proposed and foundational methods are conducted. Finally, the overall results of the proposed and baseline methods are discussed.

A. Experimental Setup

This section presents the overall experimental setup of the proposed method, including the video dataset, baseline methods, and hardware configuration information.

1) Video Dataset: The performance evaluation was carried out with 23 feature-length sports videos acquired from YouTube. Table I provides detailed descriptions of the selected videos. The videos were categorized into six distinct groups according to their content: baseball, basketball, boxing, cricket, football, and tennis. All videos used in the experiments had a resolution of 640×480 pixels. All the selected videos were examined using 10 different events selected from the action list provided in the UCF-101 dataset. The ten selected events were basketball, basketball dunk, boxing punching bag, boxing speed bag, cricket bowling, cricket shot, punch, soccer juggling, soccer penalty, and tennis swing. These events were selected based on video content. All thumbnails were selected with an accuracy exceeding 80.0% of the threshold, which was set to maintain artistic media quality. It is worth noting that the proposed method is not constrained by these events; additional events can be included based on the content present within the video.

2) Baseline Methods: This section describes the baseline methods that were compared with the proposed artistic media generation method. As explained in Section II, certain well-known approaches use entire videos to generate animated GIFs. The baseline approaches used in this study are **HECATE** [2], **HOST-ATS** [14], **AV-GIF** [44], **CL-GIF** [44], and **FB-GIF** [44].

3) Hardware Configuration: The HLS server and client hardware devices were configured locally for experimental evaluations. For HLS clients, two end-user devices were configured with different hardware configurations: a high computational resource (HCR) end-user device running on the open-source Ubuntu 18.04 LTS operating system, and a low-computational-resource (LCR) end-user machine with an NVIDIA Jetson TX2 device. The proposed and baseline approaches were set up separately on HCR and LCR machines. The HLS server machine was set up using the Windows 10 operating system and was used in the experiments. Table 2 lists the specifications of the hardware devices used in all the experiments. The hardware specifications for the LCR client resemble those of a personal computer used by most clients, with a limited memory of 8 GB.

B. Objective Evaluation

This section presents an extensive quantitative experimental evaluation of the proposed event recognition model by comparing its performance with those of prominent action recognition methods on the UCF101 dataset. In addition, the performance of the proposed method was evaluated in terms of the computational burden, network, and storage aspects using baseline methods.

1) Event Recognition: The first training and testing function of the UCF-101 dataset was used, as recommended in [52]. Each video was subsampled for up to 40 frames to train

TABLE I
THE DETAILS OF THE FEATURE-LENGTH SPORTS VIDEO USED FOR PERFORMANCE ANALYSIS OF THE PROPOSED METHOD.

S/N	Category	Title	Playtime	FPS	# Frames	# LTC	# Thumbs	YouTube ID
1	Football	Belgium vs Japan	1h 52m 14s	30	202,036	270	6734	ervkVzoFJ5w
2		Brazil vs Belgium	1h 50m 50s	30	199,506	267	6650	5OJfbYQtKtk
3		France vs Argentina	1h 50m 26s	25	165,653	266	6626	J41d0cHAfSM
4		France vs Croatia	1h 54m 1s	30	205,243	274	6841	7Fau-IwbuJc
5		Germany vs Mexico	1h 48m 56s	30	196,106	262	6536	3fYpcapas0k
6		Portugal vs Spain	1h 50m 25s	30	198,556	266	6625	Xhu5Bz1xDf0
7	Basketball	France vs USA	2h 14m 39s	30	242,135	324	8079	8YSrNfcKvA0
8		Golden State Warriors vs Brooklyn Nets	1h 40m 52s	30	181,574	243	6052	KAZ-U8vYqZg
9		Los Angeles Lakers vs Houston Rockets	1h 54m 19s	30	205,586	275	6859	aHVd9vVWVSQ
10		USA vs Spain	2h 53m 54s	25	260,886	418	10434	l9wUr-CK1Y4
11	Boxing	Canelo vs Daniel Jacobs	53m 55s	30	96,968	130	3235	1VbXe9ZjzTM
12		Davis vs Gamboa	1h 3m 2s	30	113,368	152	3782	KZtVQo8lpqY
13		Dirrell vs Davis	47m 29s	30	85,392	114	2849	sVtzzpvaEjc
14		Floyd Mayweather Jr. vs Marcos Maidana	56m 50s	25	85,259	137	3410	KYvOC7MBuUw
15	Baseball	Giants vs Dodgers	2h 11m 42s	30	236,827	317	7902	ScmHL8YVM5E
16		Giants vs Royals	2h 36m 50s	30	282,024	377	9410	YJmwofDYOeo
17		Toronto Blue Jays vs Boston Red Sox	2h 40m 50s	30	289,221	387	9650	psL-FvRg9jM
18	Cricket	India vs Pakistan	1h 25m 2s	30	153,065	205	5102	uSGCAJS6qWg
19		Peshawar Zalmi vs Islamabad United	2h 17m 15s	30	205,170	274	6845	uzErZgKuuSM
20	Tennis	Maria Sharapova vs Caroline Wozniacki	2h 10m 6s	30	233,962	313	7806	72VhC9biEFk
21		Novak Djokovic vs Daniil Medvedev	2h 1m 6s	25	181,654	291	7266	MG-RjlqyaJI
22		Novak Djokovic vs Roger Federer	4h 58m 38s	25	447,961	717	17918	TUikJi0Qhhw
23		Roger Federer vs Rafael Nadal	3h 5m 37s	25	278,448	446	11137	wZnCcqm_g-E

TABLE II
HLS SERVER AND CLIENT HARDWARE DEVICE SPECIFICATIONS.

Device	CPU	GPU	RAM
HLS Server	Intel Core i7-8700K	GeForce GTX 1080	32 GB
HCR Client	Quad-core 2.10 GHz	GeForce RTX 2080 Ti	62 GB
LCR Client	Quad ARM A57/2MB L2	Nvidia Pascal 256	8 GB

the model using the UCF-101 dataset. All images were preprocessed by cropping their central areas and resizing them to 244×244 pixels. Data augmentation was applied to reduce overfitting. A stochastic gradient descent (SGD) optimizer was used at a learning rate of 0.001 to train the model. In the experiment, an early stop mechanism was applied during the training process with patience of 30. The training data were provided in mini-batches with a size of 32, and the best validation was obtained in the 236th iteration from 1,000 iterations performed to train the sequence patterns in the data. We performed extensive experiments, adopted grid search strategies, and repeated the experiments with varying network structures, hyperparameters, and combinations to develop the FRC-GIF method. The TensorFlow toolbox was used for deep feature extraction, and a GeForce RTX 2080 Ti GPU was used for implementation. The proposed network had 88.4 million total number parameters. It is important to note that the proposed trained model solely identifies events in YouTube videos, without using the YouTube dataset for training or testing.

To the best of our knowledge, the method proposed in [17] is the only one that uses thumbnail containers to recognize events. It exhibits the best performance on the UCF-101 dataset when using thumbnail containers. The proposed thumbnail container analyzer module performs 10.37% better in terms of validation accuracy compared to the results in a previously published paper [19] and 13.12% better compared to [17]. The experimental results of the proposed and baseline approaches for the UCF-101 dataset are listed in Table III. It can be seen from the table that the proposed method outperforms most of the compared methods with a large margin.

TABLE III
COMPARISONS BETWEEN THE PROPOSED CNN ACTION RECOGNITION MODEL AND OTHER APPROACHES ON THE UCF-101 DATASET.

CNN Methods	Validation Accuracy (%)
Karpathy, Andrej, et al. 2014 [53]	65.40%
Shu, Yu, et al. 2018 [54]	76.07%
Mujtaba, et al. 2020 [17]	73.75%
Mujtaba, et.al 2022 [19]	76.5%
Ryu, et.al 2022 [18]	77.81%
Proposed	**86.87%**

TABLE IV
ABLATION STUDY.

Networks	Accuracy (%)
Xception	48.3
ResNetRS152	48.5
ConvNeXtBase	52.5
Xception + Triplet	68.12
Xception + Vortex	65.62
Xception + Vortex + Triplet	75.63
ResNetRS152 + Vortex + Triplet	75.93
ConvNeXtBase + Vortex + Triplet	82.5
ConvNeXtBase + Triplet	**86.87**

C. Ablation Study

In this subsection, we present an ablation study and comparative analysis of the modules employed in FRC-GIF and its associated network architectures. We used three different backbone network architectures: the Xception, ResNetRS152, and ConvNeXt base models. Recently, Vortex pooling methods have been extensively considered for motion summarization studies. In this regard, we also conducted experiments with vortex pooling as it aggregates frames around the target and uses an attention mechanism to select the aggregation area. The ablation results provide a means to justify the selection of the network architecture and the corresponding modules. The ablation results are presented in Table IV. As pooling and loss functions can be tested alongside backbone networks, in this regard, we compare the performance of vortex pooling and triplet loss in relation to the selected backbone network architecture, respectively. Among the network architectures, Xception and ResNetRS152 yielded similar performances, whereas ConvNeXtBase was superior to the former ones. The performance improvement using ConvNeXtBase was in agreement with the findings of [20], which performed better on several benchmarks. Therefore, ConvNeXt is selected as the primary network architecture. We also conducted experiments using vortex pooling and its combination with triplet loss. Although for Xception and ResNetRS152, it appears that the combination of vortex pooling and triplets improves the base performance, which is also the case with ConvNeXtBase, the best results have been reported using ConvNeXtBase and Triplet loss. We conjecture that an ConvNeXtBase uses the attention mechanism as part of its learning process, rendering vortex pooling redundant and potentially detrimental when combined.

1) Performance Analysis: **Static thumbnail generation.** To evaluate the performance of the proposed method, the HECATE [2] and HOST-ATS [14] methods were implemented in an HCR device and used as baseline methods with default configurations. Table V lists the number of artistic thumbnails and the computation time required (in minutes) to generate them using the proposed and baseline methods[2]. The proposed approach requires considerably less computation time than the HECATE [2] and HOST-ATS [14] methods. Significantly, all artistic thumbnails generated using the proposed method encompass personalized events. Artistic thumbnails were generated using HECATE [2] and HOST-ATS [14] as the one-size-fits-all framework. Figure 4 illustrates the artistic thumbnails produced through the proposed and baseline methods.

GIF generation on HCR device. We compared the computation time required to generate artistic animated GIFs using the proposed and baseline approaches on a HCR device. Table VI compares the computation time required (in minutes) to generate the GIFs. The HECATE method [2] analyzes every frame in a video and determines the aesthetic features that can be used to generate GIFs. The AV-GIF [44] uses the entire video and audio clips to generate animated GIFs. CL-GIF [44] uses segments and audio climax portions to generate animated GIFs. The proposed method uses considerably smaller images (thumbnails) to analyze personalized events, resulting in a significantly shorter computation time for generating animated GIFs.

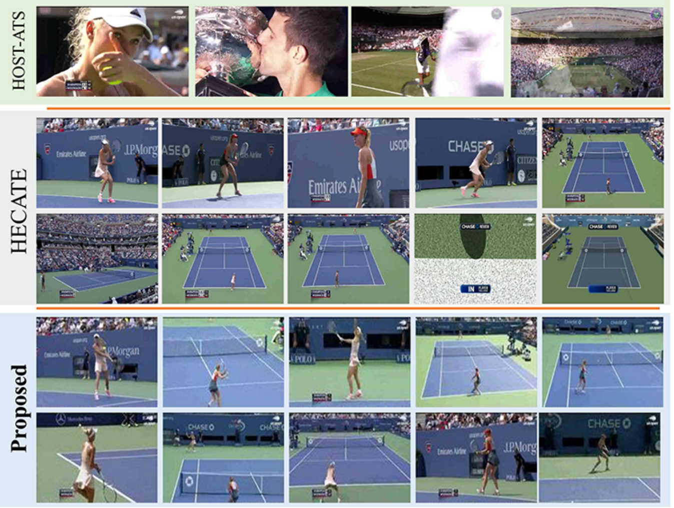

Fig. 4. Artistic thumbnails generated using the proposed and baseline methods. With default configuration of HOST-ATS [14], for every tennis video a single thumbnail is obtained.

GIF generation in LCR device. Table VI lists the computation times (in minutes) required to create artistic GIFs when implementing the baseline and proposed methods on an LCR device (i.e., NVIDIA Jetson TX2). HECATE [2], and AV-GIF [44] cannot be used in practice because they require significant computational resources as they require lengthy videos. Only the CL-GIF [44] method can be used on an LCR device to generate a GIF. The overall processing time of the proposed method was significantly shorter than that of CL-GIF [44].

Communication and storage. The HECATE [2] and HOST-ATS [14] approaches require a locally stored video file to

TABLE V
COMPUTATION TIME REQUIRED (IN SECONDS) TO GENERATE ARTISTIC THUMBNAILS USING THE BASELINE AND PROPOSED METHODS ON THE HCR DEVICE.

S/N	HECATE [2]	HOST-ATS [14]	Proposed	
	Total	Total	# Thumbnails	Total
1	3011.23	4140.096	471	**101.07**
2	5195.37	3186.32	236	**100.66**
3	2480.50	2979.346	164	**100.86**
4	3610.10	3616.829	246	**101.89**
5	2686.83	3533.306	272	**97.48**
6	4389.53	2973.497	392	**98.07**
7	7809.51	10615.848	2582	**160.36**
8	4043.81	3703.939	2935	**89.26**
9	3980.86	11150.569	3220	**102.21**
10	9530.37	6955.293	3330	**160.08**
11	882.08	2207.042	2262	**49.66**
12	1178.04	1410.047	2249	**60.97**
13	799.78	962.098	2278	**44.74**
14	842.82	3767.741	2566	**53.00**
15	5232.18	6950.231	3961	**117.28**
16	4875.19	15908.344	3992	**139.16**
17	4704.51	7142.514	5452	**141.68**
18	1517.52	5500.423	56	**75.06**
19	2922.22	7314.425	36	**98.39**
20	3911.10	4787.334	200	**115.62**
21	2064.33	2222.148	129	**106.52**
22	10696.97	9604.572	359	**263.82**
23	4491.45	5732.022	207	**224.22**

[2] With the default configuration, a single thumbnail generated using HOST-ATS [14] and ten thumbnails generated using HECATE [2] are generated.

TABLE VI
COMPUTATION TIMES REQUIRED (IN MINUTES) TO GENERATE ARTISTIC ANIMATED GIFS USING THE BASELINE AND PROPOSED METHODS ON THE HCR AND LCR DEVICES.

S/N	HECATE [2]	AV-GIF [44]	FB-GIF	CL-GIF [44]		Proposed	
	HCR				LCR	LCR	HCR
1	51.52	21.60	70.67	8.16	38.71	10.19	**2.13**
2	89.79	21.36	65.31	8.56	36.17	9.96	**2.13**
3	45.69	21.09	54.81	8.77	35.40	9.43	**2.22**
4	103.63	20.20	117.72	8.26	40.06	10.56	**2.20**
5	45.29	22.04	63.74	8.29	37.96	14.06	**2.15**
6	76.34	42.88	65.60	7.66	35.60	9.03	**2.10**
7	199.44	26.36	137.77	8.22	45.27	12.04	**3.15**
8	97.36	16.24	127.98	7.41	34.23	9.66	**1.97**
9	97.86	19.14	177.38	7.86	36.23	11.50	**2.36**
10	245.67	47.64	84.30	12.55	58.44	15.52	**3.14**
11	16.24	9.58	42.63	3.52	16.33	6.35	**1.24**
12	33.12	10.86	64.33	4.87	21.28	5.73	**1.44**
13	20.92	8.07	43.35	3.04	15.87	4.31	**1.21**
14	14.13	10.92	29.21	3.66	16.90	6.87	**1.32**
15	93.92	29.68	155.61	9.38	44.25	11.77	**2.43**
16	132.03	104.24	98.34	15.52	52.70	14.00	**2.79**
17	88.27	30.01	94.66	13.83	63.89	14.29	**2.81**
18	35.08	17.38	48.44	6.68	28.71	7.67	**1.63**
19	49.70	23.92	69.22	9.93	46.28	12.31	**2.02**
20	79.53	31.44	90.68	10.48	48.49	11.75	**2.32**
21	35.18	41.99	58.26	10.98	50.72	11.08	**2.10**
22	128.32	31.37	152.01	20.79	100.67	26.75	**4.84**
23	79.24	41.05	181.49	13.87	62.51	16.75	**4.27**

start processing. Similarly, the corresponding full-length audio file and video segment must be downloaded using the CL-GIF method to generate a GIF [44]. However, the proposed method requires only the LTC to be downloaded for the same process. For example, the video and audio sizes of the *Brazil vs. Belgium* match were 551 and 149 MB, respectively. However, the LTC size is 22.2 MB for the same video. In addition, only selected segments were obtained to generate animated GIFs using the proposed method. Thus, the proposed method significantly reduced the download time and storage requirements compared to the baseline methods.

Overall computation analysis. Table VII compares the overall computation for generating thumbnails and GIFs on the HCR and LCR devices.

To create artistic thumbnails for the 23 corresponding videos using the HCR end-user device, HECATE HECATE [2] required $1,514.27$ min, and HOST-ATS [14] 2106.06 minutes, while the proposed method required 43.37 minutes; i.e., the proposed method is 34.91 times faster than the baseline method HECATE [2] and 48.56 times faster than HOST-ATS [14] when generating the personalized artistic thumbnails. This is because HECATE [2] and HOST-ATS [14] require analysis of the entire video, whereas the proposed method utilizes LTC to create artistic thumbnails.

TABLE VII
OVERALL COMPUTATIONAL ANALYSIS (IN MINUTES) OF PROPOSED AND BASELINE METHODS.

Artistic Data	Devices	Methods	Total
Thumbnail	HCR	HECATE [2]	1,514.27
		HOST-ATS [14]	2106.06
		Proposed	**43.36**
Animated GIF	LCR	CL-GIF [44]	966.69
		Proposed	**261.63**
	HCR	HECATE [2]	1,858.25
		AV-GIF [44]	649.07
		CL-GIF [44]	212.31
		FB-GIF [44]	2,093.52
		Proposed	**53.98**

For generating the corresponding GIFs of the twenty-three feature-length videos using the HCR end-user device, the proposed method takes 53.98 minutes compared with $1,858.25$ minutes required by HECATE [2]. Again, for generating GIFs for the 23 videos on the LCR devices, the proposed method took 261.63 minutes, while the CL-GIF [44] took 966.69 minutes.

Therefore, the analysis of these 23 videos indicates that, on average, the proposed method is 34.43, 12.03, 3.93, and 38.79 times faster than the HECATE [2], AV-GIF [44], CL-GIF [44], and FB-GIF when using an HCR device, respectively. Similarly, when using the LCR device, the proposed method is 3.69 times faster than the CL-GIF [44] method. In addition, the proposed approach generates more GIFs than the baseline method. For example, whereas most methods are restricted to one GIF (e.g., AV-GIF [44], CL-GIF [44]) or a fixed number of GIFs (e.g., 10 GIFs for HECATE [2]), the proposed method can generate 25 GIFs, showing better computational efficacy than most methods in both HCR and LCR devices.

D. Subjective Evaluation

A subjective evaluation was conducted using a survey with nine participants. The participants were from three different countries, including Pakistan, Vietnam, and South Korea. A group of students was selected based on their interests in sports. The survey was based on the twenty-three videos (Table I). The quality of the GIFs was assessed using an exact rating scale. Participants were asked to grade the GIFs based on their

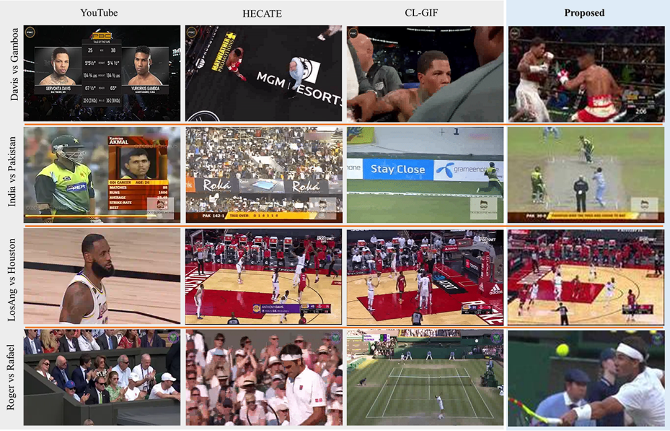

Fig. 5. Sample frames taken from GIFs generated using the proposed and baseline methods.

perceived joy. An anonymous questionnaire was designed for GIFs to prevent users from determining the method used to create a given GIF. Participants were requested to view all GIFs and rank them on a scale of 1 to 10 (1 = lowest and 10 = highest). These experiments are mostly conducted for generative studies, such as neural style transfer and image-to-image translation, where an exact quantification method for assessing user subjectivity is not available. In [55], such a subjective evaluation of DualStyleGAN was conducted. Table VIII lists the rankings of the three methods given by the participants. For the twenty-three videos, the average ratings for YouTube, HECATE [2], CL-GIF [44], and the proposed method were 5.0, 6.46, 5.65, and 7.48, respectively. The sample frames acquired from the GIFs generated using the proposed and baseline methods are shown in Fig. 5.

TABLE VIII
AVERAGE RATINGS (1∼10) ASSIGNED BY PARTICIPANTS FOR THE PROPOSED AND BASELINE METHODS.

S/N.	YouTube	HECATE	CL-GIF	Proposed
1	4.67	6.78	5.67	**8.11**
2	4.67	6.22	7	**8.56**
3	4.78	7.56	5.33	**8.44**
4	5.56	5.44	5.22	**5.78**
5	4.22	6.33	5	**7.44**
6	6.11	6.44	5.67	**6.56**
7	6.45	7.8	4.68	**8.22**
8	6.5	6.5	5.8	**8.22**
9	5.21	5.6	6.2	**7.56**
10	6.2	7.4	4.68	**8.56**
11	5.12	6.88	4.22	**7**
12	4.21	5.44	4.67	**6.56**
13	5.4	7.4	5.12	**7.88**
14	6.8	7.2	6.45	**8.44**
15	5.5	6.2	5.21	**6.86**
16	5.89	6.74	6.33	**7.44**
17	6.1	6.44	4.21	**8.11**
18	4.64	5.84	4.88	**6.44**
19	4.32	5.44	5.33	**8.44**
20	4.88	6.22	5	**8.56**
21	5.82	6.44	5.67	**7.46**
22	4.22	5.84	4.68	**6.68**
23	4.88	5.44	5.56	**7.4**

E. Discussion

The proposed method achieved significantly higher performance and a reduced computational time on both HCR and LCR devices compared to existing methods. This notable improvement stems from the proposed approach that uses LTC and video segments to generate artistic media, rather than processing the entire video. The main advantage of using LTC is that the number of thumbnails is very small compared with the number of frames in the video. For example, the *Belgium vs. Japan* football video with a duration of 1 hour 52 minutes had $202,036$ frames and 471 thumbnails. In addition, the 160×90 $(width \times height)$ thumbnails remained lightweight for every video resolution (HD, 2K, 4K, etc.) compared to the frame size of the corresponding video. The proposed method reduced the overall computational power and time required to produce artistic media on client devices.

In a qualitative experiment involving participants (described in Section IV-D), the proposed approach achieved a higher average rating than the other methods. This is primarily because GIFs are generated based on user interactions with the proposed approach. In addition, the proposed method can generate more than one GIF, which can then be used randomly to obtain a higher CTR for the corresponding video. In practical applications, the proposed method can significantly improve the CRT of newly broadcasted full-length sports videos on streaming platforms.

The proposed system can be used on a wide range of client devices with different computational resource capabilities. Owing to its simplicity and scalability in implementing multiple device configurations, it can be easily adapted to other animated image formats such as WebP and other streaming protocols. Additionally, by reducing the computational load of the servers, the proposed approach can act as a privacy protection solution by utilizing effective encryption methods [56]. Various client-based GIF generation real-time application scenarios for smartphones and set-top boxes can be considered.

V. Conclusion

In this study, we have presented a novel lightweight method which leverages a frame-ranking method for generating artistic media. The developed approach has been tailored to accommodate the limited computational resources of end-user devices. Rather than processing an entire video, the proposed method analyzes thumbnails to recognize personalized events and uses the corresponding video segments to generate artistic media. This improves computational efficiency and reduces the demand for communication and storage resources in resource-constrained devices, which is compliant with the sustainable development goals for 2030. Experimental results based on a set of 23 feature-length sports videos show that the proposed approach is 3.93 and 3.69 times faster than state-of-the-art methods used for the generation of animated GIFs when using HCR and LCR devices, respectively. Moreover, the qualitative evaluation conducted underscores that the proposed method surpassed existing methods, garnering consistently higher ratings. In the future, the proposed method will be implemented in other sports categories by considering various events using resource-constrained devices.

Acknowledgment

For open access, the author has applied a Creative Commons Attribution (CC BY) license to any Author-Accepted Manuscript version arising from this submission.

References

[1] A. Jamil, M. Abdullah, M. A. Javed, and M. S. Hassan, "Comprehensive review of challenges and technologies for big data analytics," in *2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET)*, 2018, pp. 229–233.

[2] Y. Song, M. Redi, J. Vallmitjana, and A. Jaimes, "To click or not to click: Automatic selection of beautiful thumbnails from videos," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, New York, NY, USA, 2016, p. 659–668.

[3] Y. Yuan, L. Ma, and W. Zhu, "Sentence specified dynamic video thumbnail generation," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2332–2340.

[4] H.-W. Kim, T. T. Le, and E.-S. Ryu, "360-degree video offloading using millimeter-wave communication for cyberphysical system," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 4, p. e3506, 2019.

[5] S. Bakhshi, D. A. Shamma, L. Kennedy, Y. Song, P. De Juan, and J. Kaye, "Fast, cheap, and good: Why animated gifs engage us," in *Proceedings of the 2016 chi conference on human factors in computing systems*, New York, NY, USA, 2016, pp. 575–586.

[6] S. A. Khowaja and S.-L. Lee, "Semantic image networks for human action recognition," *International Journal of Computer Vision*, vol. 128, pp. 393–419, 2020.

[7] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo, "Unsupervised extraction of video highlights via robust recurrent auto-encoders," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4633–4641.

[8] M. Sun, A. Farhadi, and S. Seitz, "Ranking domain-specific highlights by analyzing edited videos," in *European Conference on Computer Vision*, 2014, pp. 787–802.

[9] M. Redi, N. O'Hare, R. Schifanella, M. Trevisiol, and A. Jaimes, "6 seconds of sound and vision: Creativity in micro-videos," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 4272–4279.

[10] M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, and L. V. Gool, "The interestingness of images," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1633–1640.

[11] M. Gygli and M. Soleymani, "Analyzing and predicting gif interestingness," in *Proceedings of the 24th ACM international conference on Multimedia*, New York, NY, USA, 2016, pp. 122–126.

[12] M. Gygli, Y. Song, and L. Cao, "Video2gif: Automatic generation of animated gifs from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1001–1009.

[13] G. Mujtaba and E.-S. Ryu, "Human character-oriented animated gif generation framework," in *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*. IEEE, 2021, pp. 1–6.

[14] A. Husa, C. Midoglu, M. Hammou, S. A. Hicks, D. Johansen, T. Kupka, M. A. Riegler, and P. Halvorsen, "Automatic thumbnail selection for soccer videos using machine learning," in *Proceedings of the 13th ACM Multimedia Systems Conference*, 2022, pp. 73–85.

[15] T. Guardian. (2022) Netflix is reducing streaming quality amid coronavirus. how will it affect viewing in australia? [Online]. Available: https://www.theguardian.com/

[16] Y. Xu, F. Bai, Y. Shi, Q. Chen, L. Gao, K. Tian, S. Zhou, and H. Sun, "Gif thumbnails: Attract more clicks to your videos," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 3074–3082.

[17] G. Mujtaba and E.-S. Ryu, "Client-driven personalized trailer framework using thumbnail containers," *IEEE Access*, vol. 8, pp. 60 417–60 427, 2020.

[18] G. Mujtaba, A. Malik, and E.-S. Ryu, "LTC-SUM: Lightweight client-driven personalized video summarization framework using 2D CNN," *IEEE Access*, vol. 10, pp. 103 041–103 055, 2022.

[19] G. Mujtaba, J. Choi, and E.-S. Ryu, "Client-driven lightweight method to generate artistic media for feature-length sports videos," in *19th International Conference on Signal Processing and Multimedia Applications (SIGMAP), Lisbon, Portugal*, 2022, pp. 102–111.

[20] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022*, 2022, pp. 11 976–11 986.

[21] D. Misra, T. Nalamada, A. U. Aransanipalai, and Q. Hou, "Rotate to attend: Convolutional triplet attention module," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 3139–3148.

[22] Y. A. Farha and J. Gall, "Ms-tcn: Multi-stage temporal convolutional network for action segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3575–3584.

[23] S. A. Khowaja and S.-L. Lee, "Hybrid and hierarchical fusion networks: A deep cross-modal learning architecture for action recognition," *Neural Computing and Applications*, vol. 32, no. 14, pp. 10 423–10 434, 2020.

[24] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[25] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[27] K. Yang, X. Shen, P. Qiao, S. Li, D. Li, and Y. Dou, "Exploring frame segmentation networks for temporal action localization," *Journal of Visual Communication and Image Representation*, vol. 61, pp. 296–302, 2019.

[28] F. C. Heilbron, W. Barrios, V. Escorcia, and B. Ghanem, "Scc: Semantic context cascade for efficient action detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3175–3184.

[29] S. Agethen and W. H. Hsu, "Deep multi-kernel convolutional lstm networks and an attention-based mechanism for videos," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 819–829, 2019.

[30] Y. Peng, Y. Zhao, and J. Zhang, "Two-stream collaborative learning with spatial-temporal attention for video classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 773–786, 2018.

[31] J. Kim, C. Gray, P. Asente, and J. Collomosse, "Comprehensible video thumbnails," in *Computer Graphics Forum*, vol. 34. Wiley Online Library, 2015, pp. 167–177.

[32] H. Gu and V. Swaminathan, "From thumbnails to summaries-a single deep neural network to rule them all," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.

[33] D. V. Lipko, T. K. Ilyasov, and A. V. Arjakov, "Automatic generation of preview images based on video sequence analysis using computer vision," in *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*. IEEE, 2021, pp. 2154–2157.

[34] M. Rochan, M. K. K. Reddy, and Y. Wang, "Sentence guided temporal modulation for dynamic video thumbnail generation," *arXiv preprint arXiv:2008.13362*, 2020.

[35] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo, "Multi-task deep visual-semantic embedding for video thumbnail selection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3707–3715.

[36] S. Carta, A. Giuliani, L. Piano, A. S. Podda, and D. R. Recupero, "Vstar: Visual semantic thumbnails and tags revitalization," *Expert Systems with Applications*, vol. 193, p. 116375, 2022.

[37] E. Apostolidis, E. Adamantidou, V. Mezaris, and I. Patras, "Combining adversarial and reinforcement learning for video thumbnail selection," in *Proceedings of the 2021 International Conference on Multimedia Retrieval*, 2021, pp. 1–9.

[38] J. Li, S. Lin, F. Zhou, and R. Wang, "Newsthumbnail: Automatic generation of news video thumbnail," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2022, pp. 1383–1388.

[39] W. Chen, O. O. Rudovic, and R. W. Picard, "Gifgif+: Collecting emotional animated gifs with clustered multi-task learning," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, 2017, pp. 510–517.

[40] B. Jou, S. Bhattacharya, and S.-F. Chang, "Predicting viewer perceived emotions in animated gifs," in *Proceedings of the 22nd ACM international conference on Multimedia*, New York, NY, USA, 2014, pp. 213–216.

[41] J. A. Jiang, C. Fiesler, and J. R. Brubaker, "'the perfect one' understanding communication practices and challenges with animated gifs," *Proceedings of the ACM on human-computer interaction*, vol. 2, no. CSCW, pp. 1–20, Nov 2018.

[42] T. Liu, J. Wan, X. Dai, F. Liu, Q. You, and J. Luo, "Sentiment recognition for short annotated gifs using visual-textual fusion," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 1098–1110, 2020.

[43] R. Arora and Y. J. Lee, "Singan-gif: Learning a generative video model from a single gif," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1310–1319.

[44] G. Mujtaba, S. Lee, J. Kim, and E.-S. Ryu, "Client-driven animated gif generation framework using an acoustic feature," *Multimedia Tools and Applications*, 2021.

[45] Y. Pan, Y. Li, J. Luo, J. Xu, T. Yao, and T. Mei, "Auto-captions on gif: A large-scale video-sentence dataset for vision-language pre-training," *arXiv preprint arXiv:2007.02375*, 2020.

[46] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[47] FFmpeg, "Ffmpeg github page," 2023. [Online]. Available: https://github.com/FFmpeg/FFmpeg

[48] B. Fernando and S. Gould, "Discriminatively learned hierarchical rank pooling networks," *International Journal of Computer Vision*, vol. 124, no. 3, pp. 335–355, 2017.

[49] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199–222, 2004.

[50] Y. Wang, H. Huang, C. Wang, T. He, J. Wang, and M. Hoai, "Gif2video: Color dequantization and temporal interpolation of gif images," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1419–1428.

[51] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[52] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," 2012. [Online]. Available: http://arxiv.org/abs/1212.0402

[53] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[54] Y. Shu, Y. Shi, Y. Wang, Y. Zou, Q. Yuan, and Y. Tian, "Odn: Opening the deep network for open-set action recognition," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, 2018, pp. 1–6.

[55] S. Yang, L. Jiang, Z. Liu, and C. C. Loy, "Pastiche master: Exemplar-based high-resolution portrait style transfer," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 7693–7702.

[56] G. Mujtaba, M. Tahir, and M. H. Soomro, "Energy efficient data encryption techniques in smartphones," *Wireless Personal Communications*, vol. 106, no. 4, pp. 2023–2035, 2019.

Ghulam Mujtaba received B.S. in computer science from COMSATS Institute of Information and Technology, Pakistan, in 2013, the M.S. in computer science from Indus University, Pakistan in 2016, and the Ph.D. in computer engineering from Gachon University, South Korea in 2021. During his Ph.D., he was also working as a Researcher at Sungkyunkwan University (SKKU), Seoul, South Korea. His BS. degree was funded by the ICT Research and Development Scholarship by the Ministry of IT, Pakistan. He has gained vast academic and professional experience in numerous organizations. His research interests include computer vision, multimedia communications, and mobile computing.

Sunder Ali Khowaja is currently an assistant professor in the Department of Telecommunication Engineering, University of Sindh, Pakistan. He has experience of working with multinational companies as a network and RF engineer from 2008 to 2011. His research interests include data analytics, computer vision, and artificial intelligence for emerging communication technologies.

Muhammad Aslam Jarwar is a Senior Lecturer in the Department of Computing, Sheffield Hallam University, United Kingdom. He serves as a co-investigator for the Secure Ontologies for IoT Systems (SOfIoTS) project. His academic credentials include the distinction of being an Associate Fellow of the Higher Education Academy, UK, Senior Member IEEE, and a private member of the 4D Special Interest Group (4DSIG) Network. Before his current role, He garnered invaluable experience as a Research Fellow specializing in Industrial and Cyber-Physical System Security Modeling at the esteemed University College London (UCL), United Kingdom, and as a Research Associate at the University of Manchester, United Kingdom. Currently, He is the Lead Guest Editor for a special issue within the Sustainable Energy Technologies and Assessments Journal. He served as a Technical Program Committee (TPC) member for the 4th International Conference on Sustainable Technologies for Industry 4.0, showcasing his active engagement in the academic community. His impressive body of work includes authorship of numerous peer-reviewed articles and technical reports for the ITU-T. He is also recognized as a discerning peer reviewer for various esteemed journals and flagship conferences, including but not limited to IEEE Transactions on Industrial Informatics, IEEE Transactions on Network Science and Engineering, IEEE Internet of Things Journal, Springer Neural Computing and Applications, Elsevier Future Generation Computer Systems, IEEE Global Communications Conference (GLOBECOM), and IEEE International Conference on Distributed Computing Systems (ICDCS). His research interests encompass a wide spectrum, with a particular focus on areas such as the Internet of Things, Cybersecurity, Digital Twins, Ontologies, and applied AI.

Jaehyuk Choi is currently an Associate Professor in the Department of Software at Gachon University, Seongnam, Korea. He received his Ph.D. degree in Electrical Engineering and Computer Science from Seoul National University in 2008. He was with the Real-Time Computing Laboratory (RTCL) at The University of Michigan, Ann Arbor, U.S.A. as a postdoctoral researcher from Dec. 2008 to Feb. 2011. His research interest includes wireless/mobile systems, Internet of Things (IoT) Connectivity, and Sensing Systems, with emphasis on wireless LAN/PAN, mobile Wi-Fi, multipath TCP, network management, next-generation mobile networks, cognitive radios, data link layer protocols, and cross-layer approaches.

Eun-Seok Ryu is an Associate Professor at the Department of Computer Science Education in Sungkyunkwan University (SKKU), Seoul, Korea. Prior to joining Sungkyunkwan University in 2019, he was an Assistant Professor at the Department of Computer Engineering at Gachon University, Seongnam, Korea, from Mar. 2015 to Aug. 2019. He was also a Principal Engineer at Samsung Electronics, Suwon, Korea, where he led a multimedia team. He was a Staff Engineer at InterDigital Labs, San Diego, California, USA, from Jan. 2011 to Feb. 2014, where he researched and contributed to next-generation video coding standards, such as HEVC and SHVC. From Sep. 2008 to Dec. 2010, he was a Postdoctoral Research Fellow at the Georgia Centers for Advanced Telecommunications Technology (GCATT) of the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA. In 2008, he was a Research Professor at the Research Institute for Information and Communication Technology at the Korea University, Seoul, Korea. His research interests are in the area of multimedia communications, including video source coding and wireless mobile systems. He received his B.S. degree, M.S. degree, and Ph.D. in computer science from Korea University in 1999, 2001, and 2008, respectively. He is a senior member of the IEEE.