

# Client-driven Lightweight Method to Generate Artistic Media for Feature-length Sports Videos

Ghulam Mujtaba<sup>1</sup> <sup>a</sup>, Jaehyuk Choi<sup>2</sup> <sup>b</sup> and Eun-Seok Ryu<sup>3</sup> <sup>c</sup>

<sup>1</sup>*C-JeS Gulliver Studios, Seoul, Republic of Korea*

<sup>2</sup>*Department of Software, Gachon University, Seongnam, Republic of Korea*

<sup>3</sup>*Department of Computer Science Education, Sungkyunkwan University (SKKU), Republic of Korea.*  
*gmujtabakorai@gmail.com, jchoi@gachon.ac.kr, esryu@skku.edu*

Keywords: artistic media, animated GIFs, thumbnail containers, client-driven.

Abstract: This paper proposes a lightweight methodology to attract users and increase views of videos through personalized artistic media i.e., static thumbnails and animated Graphics Interchange Format (GIF) images. The proposed method analyzes lightweight thumbnail containers (LTC) using computational resources of the client device to recognize personalized events from feature-length sports videos. In addition, instead of processing the entire video, small video segments are used in order to generate artistic media. This makes our approach more computationally efficient compared to existing methods that use the entire video data. Further, 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 that are used during artistic media generation. After conducting experiments on the NVIDIA Jetson TX2, the computational complexity of our method was 3.78 times lower than that of the state-of-the-art method. To the best of our knowledge, this is the first technique that uses LTC to generate artistic media while providing lightweight and high-performance services on resource-constrained devices.

## 1 Introduction

Over the past few years, various types of streaming platforms in the form of video on demand (VOD) and 360-degree live streaming services have become very popular. In comparison to traditional cable networks that users can view on television, video streaming is ubiquitous and provides the flexibility of watching video content on various devices. In most cases, such services have very large video catalogs for users to browse and watch anytime. Yet, it is often challenging for users to find relevant content due to innumerable data and time constraints. This considerable growth has increased the need for technologies that enable users to browse the extensive and ever-growing content collections, and quickly retrieve the content of interest. The development of new techniques for generating artistic media (i.e., static thumbnails and animated Graphics Interchange Format (GIF) images) is an aspect of this demand [Song et al., 2016, Yuan

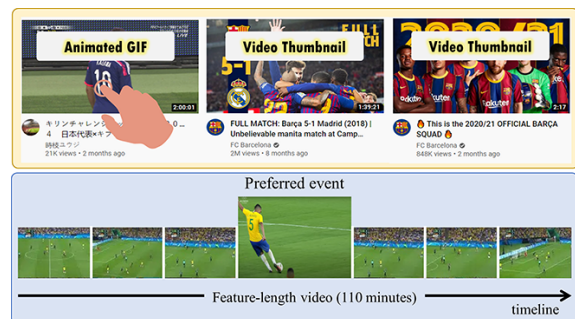





Figure 1: Artistic media in the form of static thumbnails and animated GIF images are used in popular streaming platforms to highlight recommended videos. Animated GIFs are played whenever a user hovers over the static thumbnail (above). Generally, the most preferred events are selected as static thumbnails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

et al., 2019, Xu et al., 2021].

Almost every streaming platform uses artistic media to provide a quick glimpse of video content. The static thumbnail provides viewers with a quick video preview. Meanwhile, the animated GIF delivers a condensed preview of the video for 3–15 sec [Bakhshi

<sup>a</sup>  <https://orcid.org/0000-0001-9244-5346>

<sup>b</sup>  <https://orcid.org/0000-0002-4367-3913>

<sup>c</sup>  <https://orcid.org/0000-0003-4894-6105>

et al., 2016]. Figure 1 illustrates artistic media for sports videos: (1) an animated GIF played when the user hovers the mouse on a static thumbnail (above), and (2) the most preferred frames are selected as a static thumbnail from the feature-length video. Viewers often decide whether to watch or skip a video based on its static thumbnail and animated GIF. Due to their importance, there is a growing interest in automatically creating compelling and expressive artistic media.

Click-through rate (CTR) is a significant metric for boosting the popularity of newly published feature-length sports videos on streaming platforms. However, many platforms (such as YouTube) provide only one type of artistic media for a given video without considering user preferences. In recent studies, it has been shown that personalized artistic media could play a significant role in video selection and improve the CTR of videos [Mujtaba et al., 2021]. However, creating artistic media manually is time-consuming, and their qualities are not guaranteed. Their extensive adoption and prevalence have increased the demand for methods that can automatically generate personalized artistic media from feature-length sports videos.

Nowadays, some popular video streaming sites are investigating server-side technology solutions for automatically generating personalized artistic media [Xu et al., 2021]. There are four key concerns when it comes to server-based solutions: (i) due to finite computing capabilities, personalized artistic media may not be simultaneously generated in a timely manner for multiple users, (ii) consumer privacy is prone to invasions in a personalized approach, (iii) user behavior should be overseen with recommendation algorithms, and (iv) current solutions process the entire video (frames) to generate artistic media resulting in an increase of the overall computational duration and a requirement for significant computational resources. Since personalization is one of the key elements for early media content adoption, we focused on the personalization and lightweight processing aspects of artistic media generation.

Taking the aforementioned observations into account, we propose an innovative and computationally efficient client-driven method that generates personalized artistic media for multiple users simultaneously. Considering that the computational resources are limited, we use lightweight thumbnail containers (LTC) of the corresponding feature-length sports video instead of processing the entire video (frames). Since every sports video has key events (e.g., penalty shots in soccer videos), we utilize LTC to detect events that reduce the overall processing time. Therefore, we aim to reduce the computational load and process-

ing time while generating personalized artistic media from feature-length sports videos. Twelve publicly broadcasted sports videos were analyzed to estimate the effectiveness of the proposed method. The main contributions of this research are summarized as follows:

- We propose a new lightweight client-driven technique to automatically create artistic media for feature-length sports videos. To the best of our knowledge, this is the first work in the literature to address this challenging problem.
- To support the study, we have collected twelve feature-length sports videos with approximately 1,467.2 minutes duration, in six different sports categories, namely, baseball, basketball, boxing, cricket, football, and tennis.
- We designed an effective 2D Convolutional Neural Network (CNN) model for LTC analysis that can classify personalized events.
- Extensive quantitative and qualitative analyses were conducted using feature-length sports videos. The results indicated that the computational complexity of the proposed method is 3.78 times lower than that of the state-of-the-art approach on the resource-constrained NVIDIA Jetson TX2 device (described in Section 4.2). Additionally, qualitative evaluations were conducted in collaboration with nine participants (described in Section 4.3).

The remainder of this paper is organized as follows. Section 2 provides an overview of existing literature. Section 3 describes the proposed client-driven method. Section 4 discusses the qualitative and quantitative results. Finally, the concluding remarks of this study are presented in Section 5.

## 2 Related Work

### 2.1 Animated GIF Generation Methods

Animated GIF images were first created in 1987, and have been widely used in recent years. Specifically, in the study presented in [Bakhshi et al., 2016], 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 some of the GIF features that contribute to fascinating users, such as animations, storytelling capabilities, and emotional expression. In addition, several studies [Chen et al., 2017, Jou et al., 2014] have devised methods for predicting viewers' sentiments

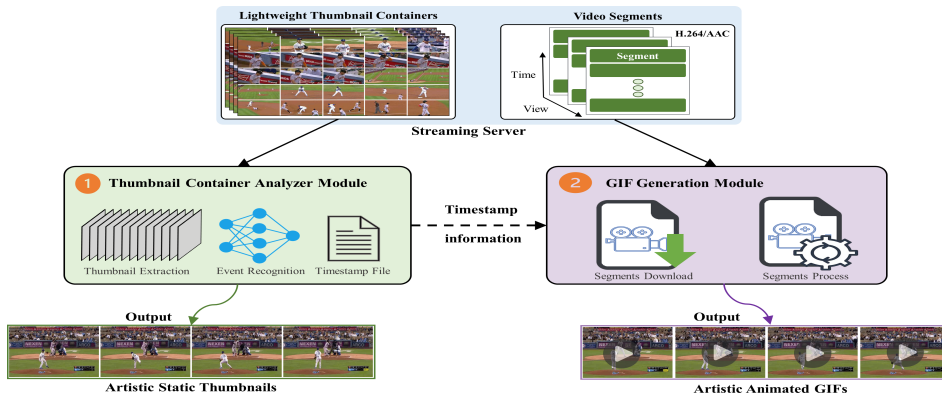


Figure 2: Semantic architecture of proposed client-driven LTC artistic media generation method. First, the thumbnail containers of the corresponding video are downloaded from the streaming server: (1) denotes the thumbnail container analyzer module that extracts thumbnails from the acquired LTC. The extracted thumbnails are analyzed using the proposed thumbnail container analyzer module to obtain artistic static thumbnails; (2) denotes the animated GIF generation module, which uses timestamp information from artistic thumbnails that are used to request and download specific video segments of the corresponding video from the streaming server. The artistic animated GIFs are generated from downloaded video segments.

towards animated GIFs. Despite the viewer engagement, in [Jiang et al., 2018], it was concluded that viewers may have diverse interpretations of animated GIFs used in communication. They predicted facial expressions, histograms, and aesthetic features; then, they compared them to the study in [Jou et al., 2014] to find the most appropriate video features for expressing useful emotions in GIFs. In another approach presented in [Liu et al., 2020], sentiment analysis was used to estimate annotated GIF text and visual emotion scores. From an aesthetic perspective, in [Song et al., 2016], frames were picked by measuring various subjective and objective metrics of video frames (such as visual quality and aesthetics) to generate GIFs. In a recent study described in [Mujtaba et al., 2021], 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 an acoustic feature to reduce the overall computational time for resource-constrained devices.

## 2.2 Video Understanding Methods

Video understanding is a prominent field in computer vision research. Action recognition [Carreira and Zisserman, 2017] and temporal action localization [Farha and Gall, 2019] are the two main issues addressed in the literature pertaining to video understanding. Action recognition involves recognizing events from a cropped video clip, which is accomplished through various methods such as two-stream networks [Simonyan and Zisserman, 2014] and recurrent neural networks (RNNs) [Donahue et al., 2015].

Another popular action recognition method uses a two-stream structure to extend a 3D CNN [Carreira and Zisserman, 2017]. It is obtained by pretraining a 2D CNN model using the ImageNet dataset [Deng et al., 2009] and extending the 2D CNN model to a 3D CNN by repeated weighting in a depth-wise manner. These features are local descriptors that are obtained using the bag-of-words method or global descriptors retrieved by CNNs. Most of the methods adopt temporal segments [Yang et al., 2019] to prune and classify videos. Recent research studies have focused on exploiting the context information to further improve event recognition. Context represents and utilizes both spatio-temporal information and attention, which helps in learning adaptive confidence scores to utilize surrounding information [Heilbron et al., 2017]. Other methods utilize time integration and motion-aware sequence learning such as long short-term memory (LSTM) [Agethen and Hsu, 2019]. Attention-based models have also been used to improve the integrated spatio-temporal information [Peng et al., 2018].

In comparison to the proposed method, HECATE [Song et al., 2016] is the most similar approach as it can generate artistic media – i.e., static thumbnails and animated GIFs. Lightweight client-driven techniques for generating artistic media are still in the early stages of development, and more effective methods are needed to bridge the semantic gap between video understanding and personalization. Additionally, most modern client devices have limited computational capabilities. Moreover, inspecting a full-length video to create artistic media is time-consuming and not reasonable for real-time solutions [Song et al., 2016].

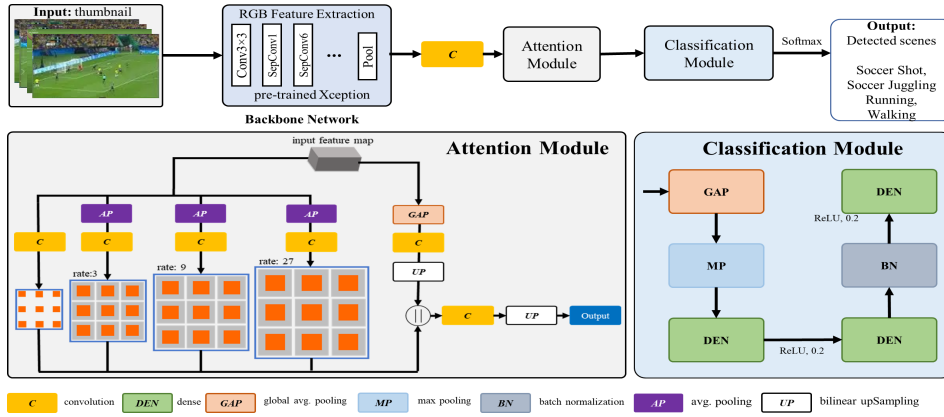


Figure 3: . Proposed architecture of 2D convolutional neural network. It analyzes each extracted thumbnail image as input. The most preferred detected events from thumbnail images are classified as output.

### 3 Proposed Method

This paper proposes new techniques for advancing research on generating anticipated artistic media using a client-driven approach. The proposed method uses LTC, which are widely used in streaming platforms for timeline manipulation of videos [Mujtaba and Ryu, 2020], instead of the entire video to analyze personalized events. Subsequently, artistic media is created within adequate processing time duration on resource-constrained client devices such as NVIDIA Jetson TX2 (i.e., an embedded AI computing device).

Figure 2 depicts the semantic architecture of the proposed artistic media method. There are two phases of generating artistic media. Each phase processes and generates a different artistic media type. In the first phase, the LTC is analyzed using the *Thumbnail Container Analyzer* module, and artistic thumbnails are obtained. The HTTP Live Streaming (HLS) server is configured to obtain the LTC [Mujtaba and Ryu, 2020]. The thumbnail containers are collected individually from the source video using FFmpeg [FFmpeg, 2020] in the streaming server. Every thumbnail container has 25 thumbnails; typically, the first frame of the video is selected as the thumbnail. The sequence of 25 thumbnails produces a single thumbnail container. Thumbnails are merged into  $5 \times 5$  containers according to playtime. A single thumbnail and thumbnail container depict the playback time of the corresponding video for 1 and 25 sec, respectively. The entire duration of the source video is covered in sequences of thumbnail containers. The size of each thumbnail and thumbnail container is fixed at  $160 \times 90$  and  $800 \times 450$  pixels, respectively (like in the work presented in [Mujtaba and Ryu, 2020]).

The timestamp information in the first phase is used to generate the artistic animated GIF from the given video segment in the second phase of the pro-

posed method. The second phase consists of the *Animated GIF Generation* module. The proposed method and its components are described in the following subsections.

#### 3.1 Thumbnail Container Analyzer Module

The thumbnail container analyzer module consists of three segments, i.e., 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., Xception model [Chollet, 2017] pre-trained on ImageNet dataset [Deng et al., 2009]) 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. Specifically, vortex pooling was used as an attention module to enhance the efficiency of the proposed neural network [Xie et al., 2018]. The module uses multi-branch convolution with dilation rates to aggregate contextual information, making it more effective. The aggregated features are then fed to a dense block-based classifier [Shen et al., 2017] to classify the events.

The thumbnail container analyzer module analyzes each extracted thumbnail individually based on the event(s) and according to user preferences. The proposed method selects a personalized artistic thumbnail from the analyzed LTC. The artistic thumbnails are selected based on a threshold that is set to maintain generated media quality. A text-based timestamp information file is generated for all selected personalized artistic thumbnails obtained from the LTC for the artistic GIF generation process. The data inside the artistic thumbnail file are ranked in chronological order.

## 3.2 Animated GIF Generation Module

The animated GIF generation module examines the segment numbers from the text-based timestamp file generated from the detected thumbnails. This information is utilized to obtain the corresponding segments from the HLS server to create an animated GIF [Mujtaba and Ryu, 2020]. The proposed method uses the first 3 sec of the segment in the animated GIF generation process. Even though the duration of all generated GIFs is fixed, it should be noted that this approach is also applicable in the case of generating a GIF with variable length. Algorithm 1 depicts the processing steps required to generate a GIF from a video with the proposed method.

---

**Data:** Input thumbnail containers  
-  $N$ : number of thumbnails  $T$  inside thumbnail containers  $LTC$   
**Initialization:**- Personalize events  $P$ ;  
Segments  $S$ ; threshold = 80  
**Main loop:** while  $i < (N)$  do  
    Extract  $T$  from  $LTC$   
    *determineEvents*( $T, P, \text{threshold}$ )  
    Identify the  $S$  number from text-based file  
    Download  $S$   
    Generate animated GIF from  $S$   
**end**  
**Function** *determineEvents* ( $T, P, \text{threshold}$ )  
    Analyze  $T$  as per  $P$   
    Select artistic  $T$  according to threshold  
    Prepare text-file of selected  $T$   
**return** text-based selected  $T$  list  
**Result:** Generated Artistic Media

---

**Algorithm 1:** Process to analyze personalize events from thumbnail containers to generate artistic media.

## 4 Results and Discussion

### 4.1 Experimental Setup

#### 4.1.1 Video Dataset

The performance evaluation was conducted using twelve feature-length sports videos obtained from the YouTube streaming platform. Table 1 provides the detailed descriptions of the selected videos. The videos are split into six categories based on their content, namely, baseball, basketball, boxing, cricket, football, and tennis. All videos used in the experiments

have a resolution of  $640 \times 480$  pixels. All selected videos were examined using ten 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 the video content. All thumbnails were selected with an accuracy exceeding 80.0% of the threshold, which was set to maintain the artistic media quality. It should be noted that the proposed method is not bound by these events; additional events can be included according to the video content.

#### 4.1.2 Baseline Methods

This section describes the baseline methods that are compared to the proposed artistic media generation method. As explained in Section 2, some well-known approaches use the entire video to generate animated GIFs. The baseline approaches are listed as follows:

- **HECATE** [Song et al., 2016] analyzes aesthetic features obtained from video frames. The corresponding video is stored locally on the device. During the process, the frames are extracted, temporarily stored, and then analyzed. HECATE [Song et al., 2016] only supports a fixed duration and number of GIFs (ten artistic thumbnails and GIFs for each video are generated for the experimental analysis).
- **AV-GIF** [Mujtaba et al., 2021] analyzes the entire audio and video files to create animated GIFs. AV-GIF generates one GIF for each video.
- **CL-GIF** [Mujtaba et al., 2021] uses acoustic features to analyze the audio climax portion and employs segments to generate GIFs. This is the state-of-the-art client-driven animated GIF generation method. Here, similar to [Mujtaba et al., 2021], only one GIF was generated using default parameters.
- **FB-GIF** Instead of analyzing the LTC, this method uses video frames of the corresponding video to detect personalized events. Initially, frames are extracted from the video; then, the proposed thumbnail container analyzer module is used to detect the corresponding events from the extracted frames.

#### 4.1.3 Hardware Configuration

The HLS server and client hardware devices were configured locally for the experimental evaluations. For HLS clients, two end user devices were configured with different hardware configurations: a high

Table 1: The details of feature-length sports video used for performance analysis in the proposed method.

S/N	Category	Title	Playtime	# Frames	# LTC	# Thumb	YouTube ID
1	Football	Belgium vs Japan	1h 52m 14s	202,036	270	6734	ervkVzoFJ5w
2		Brazil vs Belgium	1h 50m 50s	199,506	267	6650	5OJfbYQtKtk
3	Basketball	France vs USA	2h 14m 39s	242,135	324	8079	8YSrNfcKvA0
4		USA vs Spain	2h 53m 54s	260,886	418	10434	19wUr-CK1Y4
5	Boxing	Davis vs Gamboa	1h 3m 2s	113,368	152	3782	KZtVQo8lpqY
6		Dirrell vs Davis	47m 29s	85,392	114	2849	sVtzzpvaEjc
7	Baseball	Giants vs Dodgers	2h 11m 42s	236,827	317	7902	ScmHL8YVM5E
8		Giants vs Royals	2h 36m 50s	282,024	377	9410	YJmwofDYOeo
9	Cricket	India vs Pakistan	1h 25m 2s	153,065	205	5102	uSGCAJS6qWg
10		Peshawar Zalmi vs Islamabad United	2h 17m 15s	205,170	274	6845	uzErZgKuuSM
11	Tennis	Novak Djokovic vs Daniil Medvedev	2h 1m 6s	181,654	291	7266	MG-RjlqyaJI
12		Roger Federer vs Rafael Nadal	3h 5m 37s	278,448	446	11137	wZnCcqm.g-E

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 utilizing a 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 with the Windows 10 operating system and was used in our experiments. Table 2 shows the specifications of the hardware devices used in all experiments.

Table 2: 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

## 4.2 Objective Evaluation

### 4.2.1 Event Recognition

The first training/testing partition of the UCF-101 dataset was used as recommended in [Soomro et al., 2012]. Each video was subsampled up to 40 frames to train the model using the UCF-101 dataset. All images were pre-processed through cropping their central area and resizing them to  $244 \times 244$  pixels. Data augmentation was applied to reduce overfitting. The varied stochastic gradient descent optimizer was used with a learning rate of 0.01, a momentum of 0.9, and

Table 3: Performance analysis of proposed thumbnail container analyzer module.

CNN Methods	Validation Acc (%)
[Sandler et al., 2018]	59.06%
[Huang et al., 2017]	65.31%
[Karpathy et al., 2014]	65.40%
[Chollet, 2017]	68.44%
[Howard et al., 2019]	71.88%
[Mujtaba and Ryu, 2020]	73.75%
[Shu et al., 2018]	76.07%
<b>Proposed</b>	<b>76.25%</b>

the 0.001 weight decay value (SGDW) to train the model [Loshchilov and Hutter, 2017]. In the experiment, an early stop mechanism was applied during the training process with patience of ten. The training data were provided in mini-batches with a size of 32, and 1,000 iterations were performed to train the sequence patterns in the data. The Keras toolbox was used for deep feature extraction, and a GeForce RTX 2080 Ti GPU was used for the implementation. The method performed 51.32 million floating-point operations per second with a total number of 25.6 million trainable parameters.

To the best of our knowledge, the method proposed in [Mujtaba and Ryu, 2020] is the only one that uses thumbnail containers to recognize events; it had the best performance on the UCF-101 dataset when using thumbnail containers. The proposed thumbnail container analyzer module performs 2.5% better in terms of validation accuracy compared to the results in [Mujtaba and Ryu, 2020]. The experimental results of the proposed and baseline approaches on the UCF-101 dataset are listed in Table 3. All models that are used for comparison [Chollet, 2017, Sandler et al.,

2018, Howard et al., 2019, Huang et al., 2017] were trained on the UCF-101 dataset with similar configurations without adopting an attention module. It can be seen from the table that the proposed method outperforms most of the compared methods with a large margin. This is because of the context aggregation method utilized in the thumbnail container analyzer module.

#### 4.2.2 Performance Analysis

**Static Thumbnail Generation.** Static Thumbnail Generation. To evaluate the performance of the proposed method, the HECATE [Song et al., 2016] method was implemented in the HCR device and used as the baseline method with the default configuration. Table 4 shows the number of artistic thumbnails and the computation time required (in minutes) to generate them using the proposed and baseline methods. The proposed approach required considerably less computation time than the HECATE method [Song et al., 2016]. It is important to note that all artistic thumbnails obtained using the proposed method have personalized events. Meanwhile, the artistic thumbnails are generated using HECATE [Song et al., 2016] as the one-size-fits-all framework. Figure 4 illustrate artistic thumbnails generated using proposed and baseline methods.

Table 4: Computation times required (in minutes) to generate artistic thumbnails using baseline and proposed methods on the HCR device.

HECATE		Proposed	
#Thumb	Total	#Thumb	Total
10	50.19	1849	<b>1.75</b>
10	86.59	2819	<b>1.64</b>
10	130.16	2930	<b>2.01</b>
10	158.84	3376	<b>2.64</b>
10	19.63	1341	<b>0.96</b>
10	13.33	1477	<b>0.72</b>
10	14.05	2295	<b>0.86</b>



Figure 4: Artistic thumbnails generated using 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 the HCR device. Table 5 shows the comparison of computation times required (in minutes) to generate GIFs. The HECATE method [Song et al., 2016] analyzes every frame in the video and determines aesthetic features that can be used for generating GIFs. The AV-GIF [Mujtaba et al., 2021] uses the entire video and audio clips to generate animated GIFs. Meanwhile, the CL-GIF [Mujtaba et al., 2021] uses segments and audio climax portions to generate animated GIFs. The proposed method uses considerably small images (thumbnails) to analyze personalized events, which results in a significantly lower computation time for generating animated GIFs.

**GIF Generation in LCR device.** Table 5 shows the computation times required (in minutes) to create artistic GIFs when implementing the baseline and proposed methods on the LCR device (i.e., Nvidia NVIDIA Jetson TX2). HECATE [Song et al., 2016], and AV-GIF [Mujtaba et al., 2021] cannot be used in practice because they require significant computational resources owing to requiring the requirement of lengthy videos. Only the CL-GIF [Mujtaba et al., 2021] method can be used on the LCR device to generate a GIF. The overall processing time of the proposed method is significantly shorter than that of CL-GIF [Mujtaba et al., 2021].

**Communication and Storage.** The HECATE [Song et al., 2016] approach requires a locally stored video file to begin processing. Similarly, the corresponding full-length audio file and video segment must be downloaded when using the CL-GIF method to generate a GIF [Mujtaba et al., 2021]. However, the proposed method requires only LTC 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 was 22.2 MB for the same video. Thus, the proposed method significantly reduced the download time and storage requirements compared to the baseline methods.

**Overall Computation Analysis.** Table 6 shows the comparison of overall computation for generating thumbnails and GIFs on HCR and LCR devices, respectively.

To create artistic thumbnails for the seven corresponding videos using the HCR end user device, HECATE [Song et al., 2016] required 472.79 min while the proposed method required 10.57 min; i.e.,

Table 5: Computation times required (in minutes) to generate artistic animated GIFs using the baseline and proposed methods.

S/N	HECATE	AV-GIF	CL-GIF	FB-GIF	CL-GIF	Proposed	
	HCR				LCR	LCR	HCR
1	51.52	21.60	8.16	70.67	38.71	10.08	<b>2.20</b>
2	89.79	21.36	8.56	65.31	36.17	9.85	<b>2.02</b>
3	199.44	26.36	8.22	137.77	45.27	11.94	<b>2.38</b>
4	245.67	47.64	12.55	84.30	58.44	15.42	<b>3.09</b>
5	33.12	10.86	4.87	64.33	21.28	5.63	<b>1.37</b>
6	20.92	8.07	3.04	43.35	15.87	4.21	<b>1.17</b>
7	93.92	29.68	9.38	155.61	44.25	11.67	<b>2.47</b>
8	132.03	104.24	15.52	98.34	52.70	13.90	<b>2.83</b>
9	35.08	17.38	6.68	48.44	28.71	7.57	<b>1.71</b>
10	49.70	23.92	9.93	69.22	46.28	12.21	<b>2.03</b>
11	128.32	31.37	20.79	152.01	100.67	26.65	<b>4.86</b>
12	79.24	41.05	13.87	181.49	62.51	16.65	<b>3.18</b>

the proposed method is 44.72 times faster than the baseline method HECATE [Song et al., 2016] when generating the personalized artistic thumbnails. This is because HECATE [Song et al., 2016] requires the analysis of the whole video while the proposed method utilizes LTC to create artistic thumbnails.

For generating the corresponding GIFs of the twelve feature-length videos using the HCR end user device, the proposed method takes 29.31 min compared with 1,158.73 min required by HECATE [Song et al., 2016]. Again, for generating GIFs for the twelve videos on the LCR devices, the proposed method took 145.79 min while the CL-GIF [Mujtaba et al., 2021] took 550.87 min.

Table 6: Overall computational analysis (in minutes) of proposed and baseline methods.

Artistic Data	Devices	Methods	Total
Thumbnail	HCR	HECATE	472.79
		Proposed	<b>10.57</b>
Animated GIF	LCR	CL-GIF	550.87
		Proposed	<b>145.79</b>
	HCR	HECATE	1,158.73
		AV-GIF	383.54
		CL-GIF	121.58
		FB-GIF	1,170.85
		Proposed	<b>29.31</b>

Therefore, the analysis of these twelve videos indicates that, on average, the proposed method is 39.54, 13.09, 4.15, and 39.95 times faster than the HECATE [Song et al., 2016], AV-GIF [Mujtaba et al., 2021], CL-GIF [Mujtaba et al., 2021], and FB-GIF methods when using the HCR device, respectively. Similarly, when using the LCR device, the proposed method is 3.78 times faster than the CL-GIF [Mujtaba et al., 2021] method. Additionally, the proposed ap-

proach generates more GIFs than the baseline methods. For example, while most methods are restricted with one GIF (e.g., AV-GIF [Mujtaba et al., 2021], CL-GIF [Mujtaba et al., 2021]) or a fixed number of GIFs (e.g., 10 GIFs for HECATE [Song et al., 2016]), the proposed method can generate 25 GIFs, showing better computational efficacy than most of the methods in both HCR and LCR devices.

### 4.3 Subjective Evaluation

This section evaluates the subjective evaluation of generated GIFs using the proposed approach compared to those obtained from YouTube or created utilizing baseline approaches. The subjective evaluation was conducted using a survey with nine participants. Demographically, the participants were from three different countries namely Pakistan, Vietnam, and South Korea. A group of students was selected based on their interest in sports. The survey was based on the first six videos (Table 1). The quality of the created GIFs was assessed with respect to exact rating scales. The participants were asked to grade the GIFs based on perceived joy. An anonymous questionnaire was designed for the created GIFs to prevent users from determining the method used to create a given GIF. The participants were requested to view all GIFs and rank them on a scale of 1 to 10 (1 being the lowest and 10 being the highest ranking). Table 7 lists the rankings of the three methods as they were given by the participants. With regards to the six videos, the average ratings for YouTube, HECATE [Song et al., 2016], CL-GIF [Mujtaba et al., 2021], and the proposed method were 5.0, 6.46, 5.65, and 7.48, respectively. The sample frames obtained from the generated GIFs using the proposed and baseline methods are presented in Figure 5.



Table 7: Average ratings (1~10) assigned by participants for the proposed and baseline methods.

YouTube	HECATE	CL-GIF	Proposed
4.67	6.78	5.67	<b>8.11</b>
4.67	6.22	7.00	<b>8.56</b>
4.78	7.56	5.33	<b>8.44</b>
5.56	5.44	5.22	<b>5.78</b>
4.22	6.33	5.00	<b>7.44</b>
6.11	6.44	5.67	<b>6.56</b>

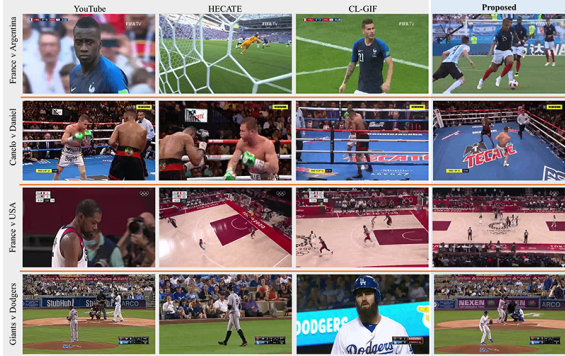


Figure 5: Sample frames taken from GIFs generated using the proposed and baseline methods.

#### 4.4 Discussion

The proposed method achieved significantly higher performance and less computation time on both HCR and LCR devices, compared with existing methods. This is because the proposed method uses LTC and video segments to generate artistic media instead of processing the entire video. The main advantage of using LTC is that the number of thumbnails is very small compared to the number of frames in the video. For example, the *Belgium vs. Japan* football video with 1h 52m duration has 202,036 frames and 6734 thumbnails. Additionally, the  $160 \times 90$  (*width*  $\times$  *height*) size of thumbnails remains lightweight for every video resolution (HD, 2K, 4K, etc.) compared to the frame size of the corresponding video. The proposed method reduces the overall computational power and time required to produce artistic media on client devices.

In the qualitative experiment involving participants (described in Section 4.3), the proposed approach obtained a higher average rating than those of other methods. This is mainly because the GIFs are generated based on user interests with the proposed approach. In addition, the proposed method can generate more than one GIF, which can then be used randomly to obtain a greater 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 plat-

forms. The client-driven approaches are in their infancy. The proposed method can be also useful for short videos [Mujtaba and Ryu, 2021].

## 5 Conclusions

This paper proposes a new lightweight method for generating artistic media using limited computational resources on end user devices. Instead of processing the 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. The experimental results that are based on a set of twelve feature-length sports videos show that the proposed approach is 4.15 and 3.78 times faster than the state-of-the-art method during the animated GIF generation process when using the HCR and LCR devices, respectively. The qualitative evaluation indicated that the proposed method outperformed the existing methods and received higher overall ratings. In the future, the proposed method could be implemented for other sports categories by considering various events using resource-constrained devices.

## ACKNOWLEDGEMENTS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) No.2020-0-00231-003, Development of Low Latency VR-AR Streaming Technology based on 5G edge cloud. This work was also supported in part by the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) with grant No. NRF-2020R1A2C1013308.

## REFERENCES

- Agethen, S. and Hsu, W. H. (2019). Deep multi-kernel convolutional lstm networks and an attention-based mechanism for videos. *IEEE Transactions on Multimedia*, 22(3):819–829.
- Bakhshi, S., Shamma, D. A., Kennedy, L., Song, Y., De Juan, P., and Kaye, J. (2016). Fast, cheap, and good: Why animated gifs engage us. In *Proceedings of the 2016 chi conference on human factors in computing systems*, pages 575–586, New York, NY, USA.
- Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset.

- In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.
- Chen, W., Rudovic, O. O., and Picard, R. W. (2017). Gifgif+: Collecting emotional animated gifs with clustered multi-task learning. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 510–517.
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255.
- Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634.
- Farha, Y. A. and Gall, J. (2019). Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3575–3584.
- FFmpeg (2020). Ffmpeg github page.
- Heilbron, F. C., Barrios, W., Escorcia, V., and Ghanem, B. (2017). Scc: Semantic context cascade for efficient action detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3175–3184.
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. (2019). Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1314–1324.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.
- Jiang, J. A., Fiesler, C., and Brubaker, J. R. (2018). ‘the perfect one’ understanding communication practices and challenges with animated gifs. *Proceedings of the ACM on human-computer interaction*, 2(CSCW):1–20.
- Jou, B., Bhattacharya, S., and Chang, S.-F. (2014). Predicting viewer perceived emotions in animated gifs. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 213–216, New York, NY, USA.
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732.
- Liu, T., Wan, J., Dai, X., Liu, F., You, Q., and Luo, J. (2020). Sentiment recognition for short annotated gifs using visual-textual fusion. *IEEE Transactions on Multimedia*, 22(4):1098–1110.
- Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization.
- Mujtaba, G., Lee, S., Kim, J., and Ryu, E.-S. (2021). Client-driven animated gif generation framework using an acoustic feature. *Multimedia Tools and Applications*.
- Mujtaba, G. and Ryu, E.-S. (2020). Client-driven personalized trailer framework using thumbnail containers. *IEEE Access*, 8:60417–60427.
- Mujtaba, G. and Ryu, E.-S. (2021). Human character-oriented animated gif generation framework. In *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, pages 1–6. IEEE.
- Peng, Y., Zhao, Y., and Zhang, J. (2018). Two-stream collaborative learning with spatial-temporal attention for video classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(3):773–786.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.
- Shen, T., Lin, G., Shen, C., and Reid, I. (2017). Learning multi-level region consistency with dense multi-label networks for semantic segmentation. *arXiv preprint arXiv:1701.07122*.
- Shu, Y., Shi, Y., Wang, Y., Zou, Y., Yuan, Q., and Tian, Y. (2018). Odn: Opening the deep network for open-set action recognition. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.
- Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27:568–576.
- Song, Y., Redi, M., Vallmitjana, J., and Jaimes, A. (2016). To click or not to click: Automatic selection of beautiful thumbnails from videos. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, page 659–668, New York, NY, USA.
- Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild.
- Xie, C.-W., Zhou, H.-Y., and Wu, J. (2018). Vortex pooling: Improving context representation in semantic segmentation.
- Xu, Y., Bai, F., Shi, Y., Chen, Q., Gao, L., Tian, K., Zhou, S., and Sun, H. (2021). Gif thumbnails: Attract more clicks to your videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3074–3082.
- Yang, K., Shen, X., Qiao, P., Li, S., Li, D., and Dou, Y. (2019). Exploring frame segmentation networks for temporal action localization. *Journal of Visual Communication and Image Representation*, 61:296–302.
- Yuan, Y., Ma, L., and Zhu, W. (2019). Sentence specified dynamic video thumbnail generation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2332–2340.