IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 5, 2020, accepted March 18, 2020, date of publication March 25, 2020, date of current version April 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982992

Client-Driven Personalized Trailer Framework Using Thumbnail Containers

GHULAM MUJTABA[1] AND EUN-SEOK RYU[2], (Senior Member, IEEE)
[1]Department of Computer Engineering, Gachon University, Seongnam 13120, South Korea
[2]Department of Computer Education, Sungkyunkwan University, Seoul 03063, South Korea

Corresponding author: Eun-Seok Ryu (esryu@skku.edu)

This work was supported in part by the Ministry of Science and ICT (MSIT) Korea, through the Information Technology Research Center (ITRC) support program supervised by the Institute for Information and communications Technology Promotion (IITP) under Grant IITP-2020-2017-0-01630.

ABSTRACT Movie trailers are prepared using a one-size-fits-all framework. These days, however, streaming platforms seek to overcome this problem and provide personalized trailers via the investigation of centralized server-side solutions. This can be achieved by analyzing personal user data, and can lead to two major issues: privacy violation and enormous demand in computational resources. This paper proposes an innovative, low-power, client-driven method to facilitate the personalized trailer generation process. It tackles the complex process of detecting personalized actions in real-time from lightweight thumbnail containers. The HTTP live streaming (HLS) server and client are locally configured to validate the proposed method. The system is designed to support a wide range of client hardware with different computational capabilities and has the flexibility to adapt to network conditions. To test the effectiveness of this method, twenty-five broadcast movies, specifically in the western and sports genres, are evaluated. To the best of our knowledge, this is the first-ever client-driven framework that uses thumbnail containers as input to facilitate the trailer generation process.

INDEX TERMS Client-driven, thumbnail containers, personalized media.

I. INTRODUCTION

Arecent study showed that every day, adults in the US spend nearly six hours watching movies [1]. This significant time allocation is still not sufficient for most of these users to explore their interests [2]. Streaming services use various recommendation systems, based on user viewing history and personal data, to suggest movies and aid quick exploration [3]. Despite these recommendations, users may require more time and interaction to identify relevance, particularly for movies and documentaries. A trailer based on user preference may help a typical user find relevant recommended movies/documentaries.

Typically, a trailer can be regarded as a type of video in summary, as it is a short version of the original movie [4]. Trailers are produced to highlight the movie and attract interested patrons [5], [6]. Apart from that, creating trailers usually involves high levels of cognitive effort due to their diversity, even for similar movies [7]. Therefore, a small number of trailers are generated for movie-making process [8]. In some cases, only one trailer is generated, hence everyone gets the same trailer for the movie. Because each user may cast diverse interests, even for the same movie, he/she may not show interest in a trailer that is not related to their interest [9].

In order to automate personalized movie trailers, centralized server-side solutions are under investigation and development [10], [11]. The advantage of centralized servers is that all information, including user personal data and functionalities, are located in the same place. Their disadvantage is that clients need to send data concerning every user interaction to servers. Thus, user privacy can be violated. Moreover, with the increasing number of users and need for interactive interfaces, the amount of information transmitted between clients and servers can grow rapidly. Thus, servers need enormous computational resources to process all the user information. In addition, the recommender systems explicitly control and affect the behavior of users [12]. These factors prompted us to research the conversational method where users can control system according to their current mood and interests. This also led us to explore the feasibility of a client-side approach

The associate editor coordinating the review of this manuscript and approving it for publication was Yin Zhang.

which can generate personalized trailers according to users' interests, with low computational resource demand.

The responsiveness of the server-based personalized trailer system can be difficult to maintain in real-time when there is a large number of concurrent users. This paper proposes a real-time and low-power client-driven method to facilitate the personalized trailer generation process. It tackles the complex process of detecting personalized actions in real-time from lightweight thumbnail containers. The cross-platform HLS server and client are locally configured to validate the proposed method. Instead of open/close connections for every request/response for thumbnail containers and video segments, the HTTP persistent broadcast connection adapts to reduce computational resources. To test the effectiveness of the method, the personalized actions in 25 broadcast movies, in the western and sports genres, were identified and evaluated, and subjective evaluation was conducted to compare the number of relevant actions available in official and generated trailers. The proposed approach requires minimum computational resources and can provide comprehensive, privacy protection solutions [13]. Every user can get distinct trailers simultaneously, according to their interests, as shown in the conceptual diagram in Fig.1.

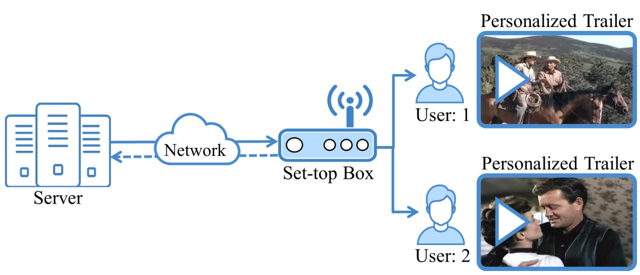

FIGURE 1. **Conceptual diagram of the proposed client-driven personalized trailers framework.**

Compared with existing methods, the main contributions of this paper are concluded thus: (1) We propose a first-ever thumbnail container-based client-driven framework to detect personalized actions in order to expedite and facilitate the trailer generation process. (2) The proposed system adopts an HTTP persistent broadcast connection to reduce the corresponding responses and network bandwidth. (3) The HLS server and client are designed locally to validate the effectiveness of the proposed method.

The remainder of the paper is structured as follows. Section II provides a brief summary of related work. Section III discusses the proposed thumbnail-based trailer framework approach. A detailed implementation of thumbnail containers for personalized trailers is discussed in Section IV. Section V presents experimental results along with discussions and Section VI concludes this study.

II. RELATED WORK

A trailer can be quite cinematic, with its own background music, sophisticated shot transition, and post-produced features such as overlaid text [14]. Due to the artistic and creative aspects, a specific field to facilitate the movie trailer generation process did not generally receive the same attention as movie summarization and highlights. Among the few works addressing the specific issue, the work by [15] is the first-ever to introduce a half-automated method to generate a movie trailer through human-AI collaboration. The study specifically interpreted the different types of emotions in horror movies and selected the ten best moments from a full-length feature movie to create a movie trailer. Another approach proposed in [14] focused on action movies. It specifically studied the visual motion levels of the entire movie and detected specific audio cues (e.g., speech, music, silence, etc.) to define and select individual sequences for creating a trailer. In [16], an approach in the context of a TV program focused on identifying textual correspondences between the sentences in the program summary (provided by the Electronic Program Guide) and closed captions from the original video. In order to create the trailer, researchers first prepared a textual summary.

TABLE 1. **Comparison of our proposed thumbnail-based approach with existing work; A: *Audio*, V: *Video*, Tx: *Text*, TC: *Thumbnail containers*.**

Method	Genre	Video type	Personalize	Media content			
				A	V	Tx	TC
Smeaton,et al [14]	Action	Movie	×	✓	✓	×	×
Smith,et al [15]	Horror	Movie	×	✓	✓	×	×
Kawai,et al [16]	Adventure	Documentary	×	×	×	✓	×
Proposed	Western, Sports	Movie, Documentary	✓	×	×	×	✓

Movie summarization, in particular video summarization techniques have attracted much research interest, and many works have been proposed in last decades [17]–[20], [21], [22]. In recent years, personalized video summarization by extracting keyframes from videos has been enthusiastically investigated in diverse applications [23]–[25]. Beyond visual features to extract keyframes, several other features are considered in recent work, including viewer attention [26]–[28], audio signal [29], and subtitles [30]. Moreover, semantic information has also been exploited to summarize videos, including special events [31], [32], and storylines [33]. More recently, a personalized keyframe recommendation framework using time-synchronized comment information was proposed [34]. Recently, several approaches have focused on bringing high-level semantics into video summarization, such as events [35], and objects [36]. In a different approach, important human faces and objects are used to summarize a video [37]. In addition, video highlight generation is also extensively investigated for the first person [38] and sports [39].

Most of the described trailer generation methods ignore the cost of computational resources, processing time and user interest. Table 1 shows a comparison of proposed and previous movie trailer generation methods for broadcast movies

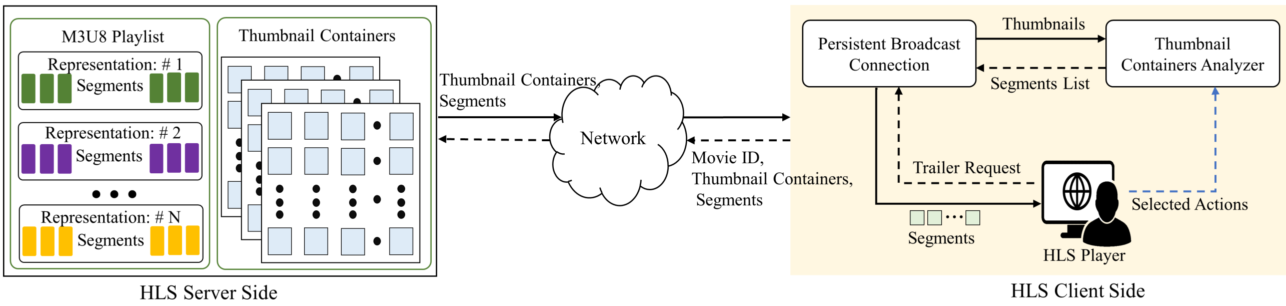

FIGURE 2. **The high-level architecture of the proposed thumbnail-based trailers framework.**

in specific genre video types (movies/documentary). Instead of entire movies, the proposed method analyzed lightweight thumbnail containers for the trailer generation process. The entire process was performed using client device computational resources that will minimize server-side computations.

III. PROPOSED THUMBNAIL-BASED TRAILERS FRAMEWORK

In recent years, several methods have been designed to improve the demanding timeline manipulation for video browsing [40]–[42]. One of the most popular methods for web-based video players is the use of lightweight thumbnail containers. Users can skim through the video player for efficiency and instantly preview a lengthy video, by watching the thumbnails preview in the web player interface. As the thumbnails are easy to integrate and navigate in web-based video players, several famous video-on-demand (VoD) streaming platforms adopt these in their end-user video players. As an example, Fig. 3 shows a thumbnail container of the Vengeance Valley movie received in the client web-based video player (left), and a thumbnail previewing a particular duration (right). It is observed that a strong, vibrant, and relevant thumbnail draws more attention by giving the user an easy quick preview [43].

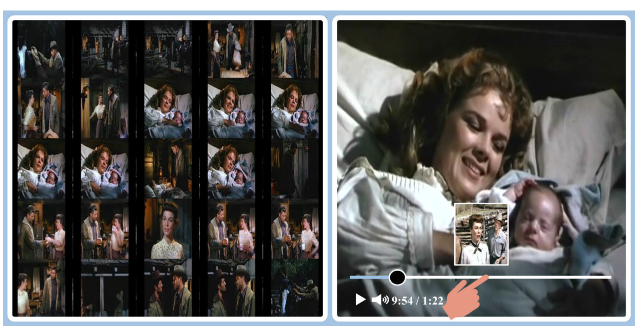

FIGURE 3. **Orientation of thumbnails on a single thumbnail container image in YouTube (left) and the thumbnail usage for instant preview on the web-based video player (right).**

The core idea of the proposed framework is to facilitate personalized trailer production in real-time by using mini- mum client device computational resources. Instead of analyzing an entire movie, it uses thumbnail containers to find out relevant action/s. It is designed to support a wide variety of client machines that have different computational capabilities and adjust dynamically in different network conditions. Fig 2 shows the system architecture of the proposed thumbnail-based framework. The system consists of two major parts (i.e., the HLS server and the client). The proposed framework mainly focuses on the client-side application and an entire trailer generation process is performed using client device resources. From the server-side implementation perspective, the cross-platform HLS server is locally configured.

A. CROSS PLATFORM HLS SERVER

The HLS server is designed to enable multiple users download thumbnail containers and segments concurrently on heterogeneous devices. For this purpose, the Microsoft Windows internet information services (IIS) web server is configured locally. IIS supports most network protocols such as HTTP, HTTP/2, HTTPS, FTP, FTPS, SMTP, and NNTP [44]. Initially, the entire movie is encoded as H.264/AAC MPEG-2 transport streams (.ts) segments in multiple bitrates using [45]. The MPEG-2 transport stream, which consists of 188-byte fixed-length packets [46], is suitable for transmissions when there is a potential corruption or loss of packets. Each segment contains a ten-second playback portion of the movie with a continuous timestamp. The list of the segments in their playback order is stored in a text-based playlist file (m3u8). Each individual bitrate playlist contains URLs pointing to movie segment files [47].

The dedicated HLS server also contains thumbnail containers, which are extracted separately from the original movie. The first frame is separately extracted in each second from the original movie using [45]. Then, the extracted frame resized into thumbnail size which is 160×90 (i.e. *width* × *height*). The extracted thumbnails then merge into 5×5 on multiple thumbnail containers. Each thumbnail presents a one-second duration, and one thumbnail container represents 25 seconds duration of the movie. The size of thumbnail and orientation of thumbnail containers were adopted after investigating the current YouTube web-based video player. These thumbnail containers are associated with the original movie and cover its entire length. Algorithm 1 presents the thumbnail container generation process from MPEG video.

Algorithm 1 Thumbnail Containers Generating Mechanism From MPEG Video

Data: Input MPEG video
- N: number of frames inside MPEG
Initialization:- Thumbnail size *width* × *height*; Interval between frames in seconds (*interval*) = 1 second; Thumbnail container orientation (*orientation*) *row* × *column*; Thumbnail counter (*count*) = 0
Main loop: **while** $i < (N)$ **do**
    Determine count = duration of MPEG video / interval
    while $j < (count)$ **do**
        Extract frame after interval
        Resize and save as thumbnail in temp location
    end
end
Function *generateContainer (thumbnail, orientation)*
    Place thumbnails as per orientation
    Save thumbnail container n_i
Result: Set of Thumbnail Container

B. HTTP PERSISTENT CONNECTION

Several requests will be initiated from the client to IIS server to collect the thumbnail containers and segments in the process to generate personalized trailers. The cost-effective HTTP 2.0 persistent connection is used because it enables the exchange of numerous requests and returns of data simultaneously, over a single TCP connection [48]. The advantage of the persistent connection communication channel is that it remains open for HTTP requests and responses rather than closing after a single exchange. Fig. 4 shows the HTTP persistent network transmission between client and server sides. The performance of the persistent connection adaptive streaming was evaluated in [49]. An open connection is faster for frequent data exchanges. Using a persistent connection has several advantages: for example, the overall CPU usage and round trips are reduced because of fewer new connections and TLS handshakes [50]. In addition, communication overheads are saved by leaving a connection open rather than opening and closing sessions for each request.

C. THUMBNAIL-BASED HLS CLIENT

The client-side consists of two major components: a deep learning-based action recognition model, and a web-based HLS video player.

1) ACTION RECOGNITION

The deep-learning action recognition model is trained to recognize actions from thumbnail containers. We computed the embedding using pre-trained models on the UCF-101 action dataset [51]. UCF-101 is an action recognition dataset of realistic action videos collected from YouTube, that consists of 101 action categories. It has 13,320 videos from 101 diverse action categories. The videos of this dataset are

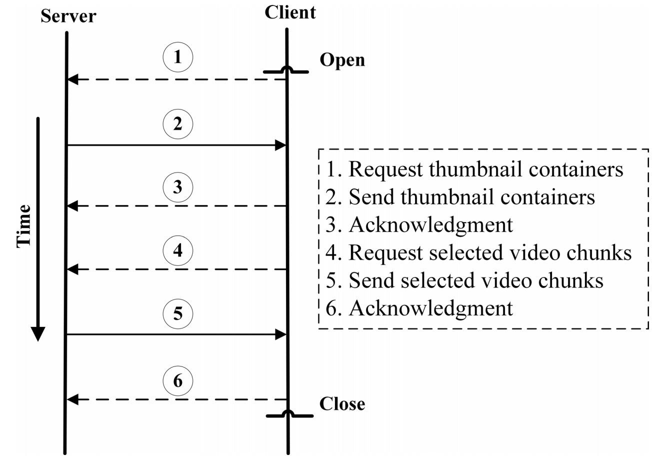

FIGURE 4. HTTP persistent connection communication between client and server.

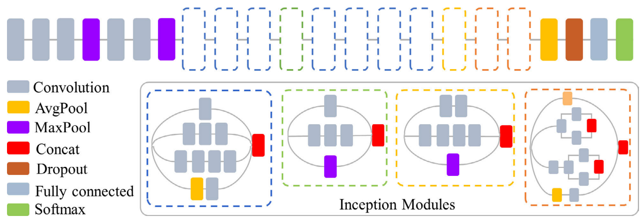

FIGURE 5. Convolutional neural network Inception-v3 architecture.

challenging and include large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, and illumination conditions. To train the UCF-101 dataset, each video is subsampled down to 40 frames. The frame-level features are extracted using the state-of-the-art Inception-v3 image annotation model, pre-trained on ImageNet [52]. We fine-tuned and trained the model freezing the top layers of the network and updating the weights in only the final. The model is trained using Stochastic Gradient Descent (SGD) optimization algorithm with 0.01 learning rate. Fine-tuned Inception-v3 trained on UCF-101 dataset obtained training accuracy of 91.41% and testing accuracy was 73.75% with 24,004,485 parameters. The first train/test split of the dataset suggested by the reference website is used. The architecture of the Inception-v3 convolutional neural network is shown in Fig. 5.

2) WEB-BASED HLS VIDEO PLAYER

The interface of the HLS video player is designed and integrated into an open-source JavaScript library [53]. It relies on HTML5 video and media source extensions for playback by transmitting an MPEG-2 transport stream into ISOBMFF fragments. The data delivery is entirely client-driven, which means that the video player can determine when to request each segment from the playlist file in the playback order or with a specific timestamp. Moreover, the HLS video player can shift between the different bitrates

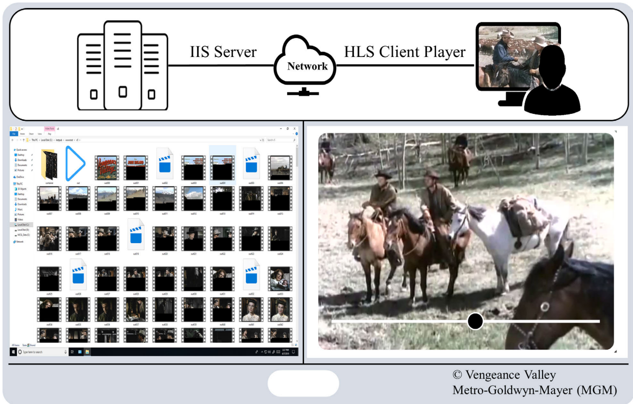

FIGURE 6. IIS server containing segments and thumbnail containers (left side), web-based HLS client video player interface displaying generated trailer (right side).

of a movie during playback, as the HLS server encodes a movie into multiple bitrates.

Fig. 6 shows the IIS server containing the segments and thumbnail containers, and the web-based video player interface on the client-side. The proposed framework supports the HLS protocol for VoD sessions. Media contents (e.g., segments, playlists, etc.) representing the entire duration can be assessed in the VoD session on the client-side using a web-based HLS video player. The list of movie segments in their playback order is stored in the text-based m3u8 playlist file to display the movie sequentially in the player. The HLS video player uses the existing m3u8 playlist to determine the available bitrates and location of the movie segments in the server.

IV. IMPLEMENTATION

This section provides a detailed description of the proposed, client-driven thumbnail-based framework for personalized trailers. In order to validate the proposed approach, twenty-five movies were analyzed with different playtime. Table 2 shows all the movie titles, their release years, genres, IMDB ratings, and duration. We analyzed specifically sports and western genres including cinematographic movies and documentaries. Since a movie may consist of more than one genre, the most dominant genre of the movie is given (western or sport). Every movie is different, but there are commonalities between movies that belong to the same genre. As an example, typical horse-riding actions are similar in every western genre movie. This helped in creating trailers by generalizing actions categorized according to movie genres. All the movies were examined by six distinct actions that were selected from the UCF-101 action categories list. Fig 7 shows the list of all the actions that are used to analyze movies.

In order to analyze the thumbnail containers, the client-side requests thumbnail containers for a specific movie. The HTTP server simply provides the pre-generated thumbnail containers using an HTTP persistent connection. The transferred thumbnail containers cover the entire length

FIGURE 7. First four images display the sample of actions selected for sports genre movies: cricket bowling, cricket shot, soccer shot, and soccer juggling; the last two images display the sample of actions selected for western genre movies: a horse race, and horse riding.

TABLE 2. The list of movie titles used for analysis in the entire client-driven process.

No	Title and release year	Genre	IMDB	Duration
1	89 (2017)	Sport	7.8	1 h 31 min
2	Bobby (2016)	Sport	7.1	1 h 37min
3	Bodyline E-1 (1984)	Sport	8.5	49 min
4	Bone Tomahawk (2015)	Western	7.1	2 h 12 min
5	Dakota (1945)	Western	6.1	1 h 22 min
6	Django (1966)	Western	7.3	1 h 31 min
7	Django Unchained (2012)	Western	8.4	2 h 45 min
8	Goal! (2005)	Sport	6.7	1 h 58 min
9	Iqbal (2005)	Sport	8.1	2 h 12 min
10	Kenny (2017)	Sport	7.3	1 h 26 min
11	Lagaan (2001)	Sport	8.2	3 h 44 min
12	Little Big Man (1970)	Western	7.6	2 h 19 min
13	M.S. Dhoni (2016)	Sport	7.7	3 h 4 min
14	Oklahoma! (1955)	Western	7	2 h 25 min
15	Pelé (2016)	Sport	7.2	1 h 47 min
16	Playing Away (1987)	Sport	6.6	1 h 40 min
17	Shanghai Noon (2000)	Western	6.5	1 h 50 min
18	Take The Ball, Pass The Ball (2018)	Sport	8.2	1 h 49 min
19	The Game of Their Lives (2005)	Sport	6.1	1 h 41 min
20	The Indian Fighter (1955)	Western	6.4	1 h 28 min
21	The Rider (2017)	Western	7.4	1 h 44 min
22	The Tracker (2002)	Western	7.4	1 h 30 min
23	The Train Robbers (1973)	Western	6.5	1 h 32 min
24	True Grit (2010)	Western	7.6	1 h 50 min
25	Vengeance Valley (1951)	Western	5.9	1 h 23 min

of the movie. Table 3 shows the movie's frame per second (FPS), total frames in a movie, thumbnail containers, and thumbnails. The number of frames in a movie is associated with FPS. Meanwhile, the number of thumbnail containers is associated with the duration of a movie. A very low bitrate is required to transmit the thumbnail containers due to their very small number and lightweight as compared to the total frames of the movie.

The proposed system uses a canvas to separately extract each thumbnail from the downloaded thumbnail containers in the client device using [45]. A single thumbnail container contains 25 thumbnails. The trained deep-learning action recognition model gets two inputs every time to recognize actions: preferred actions and a thumbnail image. The user selects action/s for the movie using web-interface.

TABLE 3. Total number of frames in a movie, thumbnail containers, and thumbnails.

No	FPS	Total frames	Thumbnail containers	Thumbnails
1	25	135300	217	5412
2	25	1420204	225	5608
3	25	73820	119	2952
4	23	189983	317	7923
5	24	117895	197	4917
6	24	131749	220	5495
7	24	237909	397	9922
8	23	169932	284	7087
9	25	189945	304	7597
10	25	129649	208	5186
11	23	322944	539	13469
12	24	200509	335	8362
13	24	265933	444	11080
14	24	201422	337	8401
15	23	147545	257	6415
16	24	146237	244	6093
17	23	158648	265	6617
18	25	163972	263	6558
19	24	145699	244	6076
20	23	127126	213	5302
21	24	148404	248	6189
22	24	141143	236	5886
23	23	132192	221	5514
24	24	158733	265	6620
25	30	147448	197	4919

Algorithm 2 Personalize Action/s Selection Process for the Movie

Data: Input movie
- N: number of action/s; User preference
Initialization:- Action
Function *updatePreference (Action, Preference)*
    Select *Action*
    if *Action present in preference* **then**
        continue
    else
        Add *Action* in *preference*
    end
    Save and update *preference*
Result: Personalize action/s

Algorithm 2 presents action selection process for the movie. If the action is already present in user preference, the system will use that action. In a case if that selected action is not present then, the system adds that action in user preference. User can use more than one actions for the personalized trailer process. Fig. 8 illustrates the thumbnail analyzer process. The deep- learning model analyzes each thumbnail separately, according to a selected action(s) by the user. Once all the thumbnails are analyzed, the system will rank them according to their accuracy. The system selects only thumbnails with high accuracy to provide the best user experience. The selected and ranked thumbnails are stored in the text-based list.

The detected thumbnails list is analyzed to create an additional-segments text-based list to request specific segments with different timestamps from the server.

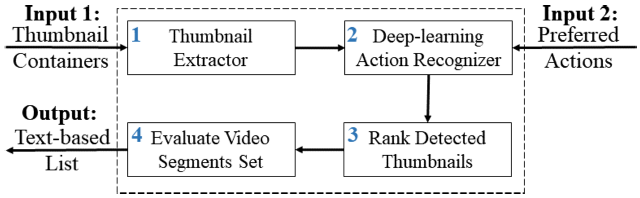

FIGURE 8. Thumbnail containers action analyzer process.

Algorithm 3 Request Video Segments From Input Detected Thumbnail Containers List

Data: Input thumbnails list
- N: number of detected thumbnails
Initialization:- Thumbnail container TC; Thumbnail T; Segment number $(segNo) = 0$
Main loop: **while** $i < (Max\ of\ N)$ **do**
    Determine detected TC number
    Determine T number
    Determine segNo
    if $segNo > 1$ **then**
        Save segNo into text-based list
    end
end
Result: Text-based Segments List

Algorithm 3 shows the process to produce a segments list from the detected thumbnails list. Subsequently, the client-device requests specific segments with a different timestamp using a text-based list. The HTTP server sends the segments at a time upon a segment request from the client. The client-side aggregate segments so that the movie can be presented to the user in a continuous stream. If a segment takes too long to download, an alternate bitrate is selected and the m3u8 playlist for that bitrate is consulted.

It is observed during the trailer generation process sometimes, that some pictures (frames) are not relevant to the selected actions in a segment. The video player has to decode the whole segment even though it contains some relevant pictures. Which is why enormous computational resources are required to decode and encode the segment. It is possible, with the entry points and exit points (timestamp), to get only relevant pictures in a segment, as shown in Fig. 9. The video player (including decoder) point of view and subset of segment (also called partial segment) can be supported. The decoder may decode all video frames in a segment for intra picture decoding but displays the video frames according to the entry point and exit point information. By adapting timestamp information at decoder side, no heavy video processing will be required.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. EXPERIMENTS

We carried out productivity and subjective experiments to empirically evaluate the proposed method. Our dataset

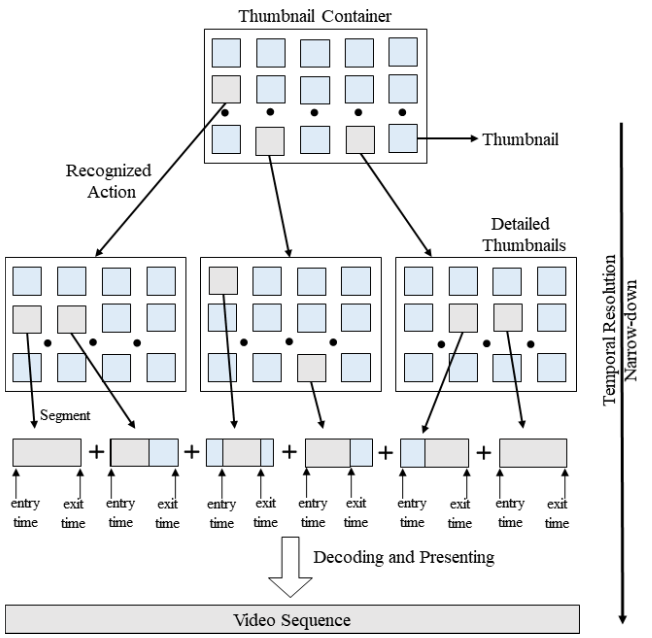

FIGURE 9. Preparation of personalized video sequence.

TABLE 4. Performance comparison of thumbnails with number of frames in movies.

No	FPS	Total frames	Thumbnails	Percent
4	23	189983	7923	**4.17**
18	25	163972	6558	**3.99**
25	30	147448	4919	**3.33**

consisted of 25 publicly broadcast, available cinematographic movies and documentaries. 12 sports and 13 western genres. We also collected official trailers of the listed movies. All movies and their trailers had frame size 640 × 480. Table 9 shows the names and number of trailers, YouTube video ID, and number of views. If a movie had more than one official trailer, only the one with the highest views was listed and analyzed. The YouTube streaming platform was used to search for the official trailers for all listed movies. "Trailer" and "promo" were the main keywords used in the search process for official trailers. Additionally, no trailer could be identified for two of the listed movies (Bodyline, and Playing Away) in our dataset. Thus, it is assumed that there is no single official trailer available online for these two listed movies.

The proposed method analyzes a very small number of lightweight thumbnail images compared to the number of frames in a movie. Table 4 shows the percentage of lightweight thumbnails used in the trailer generation process in comparison to the number of frames in the movie. Because the thumbnail size is very small (160×90), it requires less computational resources and processing time. The generated trailer duration is less than 150 seconds, which is the maximum length approved by the motion picture association of America (MPAA) [8]. The trailer duration is maintained by dynamically ranked thumbnails during the action recognition process.

TABLE 5. Time required to process the entire movie and proposed thumbnail-based approach.

No	Entire movie		Total (minutes)	Proposed (minutes)
	Frame extraction	Action recognition		
4	14.66	94.24	108.9	**2.85**
18	16.39	98.39	114.78	**2.55**
25	4.07	52.2	56.27	**2.04**

Table 5 shows the comparison of processing time, in minutes, of the entire movie and the proposed thumbnail-based approach. In order to analyze the entire movie, two phases were completed: frame extraction from the original movie and action recognition from extracted frames. The frame extraction process was achieved using [45], and action recognition using III-C.1. Meanwhile, all seven steps achieved that described in Section IV in the proposed thumbnail-based approach. All seven steps include requesting thumbnail containers for a specific movie, thumbnail extraction, recognizing actions from thumbnails, ranking detected thumbnails, preparation of segment list, requesting segments and aggregating segments. Even though more steps were used in the thumbnail-based approach, the overall time is tremendously low compared to processing the entire movie. For this comparison, the client machine configured with dual quad-core 2.10 GHz Xeon processors, GeForce RTX 2080 Ti, 62 GB RAM, and running on open-source operating system Ubuntu 18.04 LTS was used. Table 6 shows time required to download multiple thumbnail containers resolutions with different RAM sizes. This paper uses only one thumbnail containers resolution (800×450) for all movies in entire process.

A subjective evaluation was conducted to get the user perception of the proposed approach. A group audience of 46 participants performed in the test viewing process. We conducted the evaluation by dividing the number of participants into two different groups. The groups were divided based on their interest in movie genre (sports or western). In terms of demographics, the participants in the survey covered a wide range of ages from 18 to 40 years and 12 different geographic locations. Gachon University's students and professors participated in the subjective analysis.

The trailer may have their own background music, and post-produced features such as overlaid text as described in Section II. Furthermore, for the trailer the most exciting and pertinent scenes are selected from the movie to tell the story in a highly condensed fashion, which is essential to sustain the viewer intrigue and excitement. In current version of the proposed thumbnail-based personalized trailer, it lacks background music, post-production features and the most pertinent scenes selection that can engage the user for entire time and arouse curiosity. Due to these limitations, we were unable to perform users' evaluations to rank original and generated trailer. Thus, a purely statistical analysis was conducted to find the number of actions according to participant interest. In addition, participants also took part

TABLE 6. Time required to download thumbnail containers.

No	RAM	Thumbnails size	Thumbnails containers size	Duration (seconds)
4	16 GB	160 × 90	800 × 450	2.35
		320 × 180	1600 × 900	3.31
		480 × 270	2400 × 1350	4.25
	32 GB	160 × 90	800 × 450	2.36
		320 × 180	1600 × 900	3.23
		480 × 270	2400 × 1350	4.38
	64 GB	160 × 90	800 × 450	2.06
		320 × 180	1600 × 900	3.1
		480 × 270	2400 × 1350	4.23
18	16 GB	160 × 90	800 × 450	2.01
		320 × 180	1600 × 900	2.88
		480 × 270	2400 × 1350	3.85
	32 GB	160 × 90	800 × 450	2.01
		320 × 180	1600 × 900	2.74
		480 × 270	2400 × 1350	3.72
	64 GB	160 × 90	800 × 450	1.79
		320 × 180	1600 × 900	2.61
		480 × 270	2400 × 1350	3.53
25	16 GB	160 × 90	800 × 450	1.59
		320 × 180	1600 × 900	2.38
		480 × 270	2400 × 1350	3.04
	32 GB	160 × 90	800 × 450	1.57
		320 × 180	1600 × 900	2.25
		480 × 270	2400 × 1350	3.03
	64 GB	160 × 90	800 × 450	1.39
		320 × 180	1600 × 900	2.05
		480 × 270	2400 × 1350	2.87

TABLE 7. Recognized actions from official and generated trailers in sports genre movies.

No	Official trailers		Generated trailers	
	Duration	Actions	Duration	Actions
2	2 min 13 sec	3	1 min 53 sec	**6**
8	2 min 20 sec	5	1 min 33 sec	**7**
13	3 min 17 sec	6	1 min 43 sec	**8**

TABLE 8. Recognized actions from official and generated trailers in western genre movies.

No	Official trailers		Generated trailers	
	Duration	Actions	Duration	Actions
12	4 min 20 sec	2	1 min 49 sec	**6**
23	3 min 32 sec	3	2 min 11 sec	**8**
25	2 min 34 sec	3	1 min 50 sec	**6**

in noting similar actions in both the official and generated trailers.

We concurrently showed the participants the official and generated trailers. The result of the subjective analysis is displayed in two sequences. Each sequence contains three movies from each movie genre. Table 7 shows the results of recognized actions from sports genre movies, which include movie id, duration of official and generated trailers, recognized personalized actions from the generated and official trailers. Participants count the number of actions according to the movie genre as defined in Section IV. Table 8 shows the results of recognized actions from western genre movies. The results show that the generated trailers have a larger number of actions as compared to official trailers.

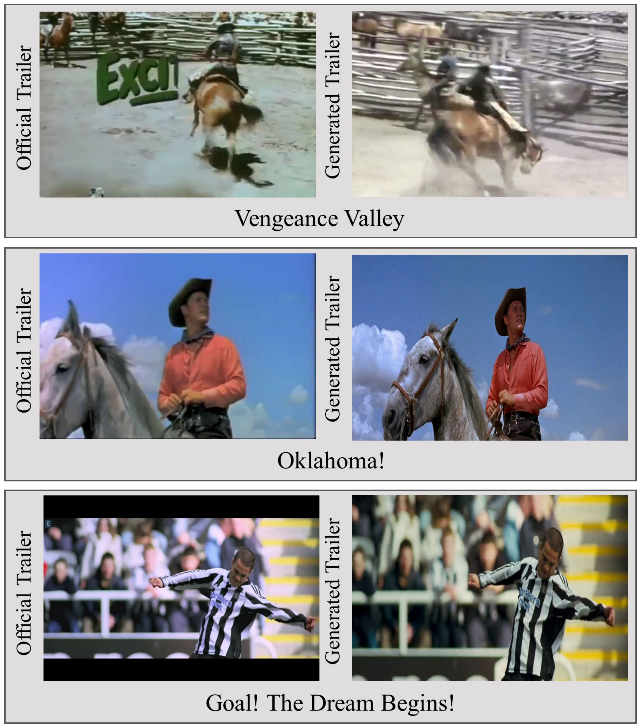

FIGURE 10. Identical actions in compression with broadcast official and generated trailers.

During the subjective evaluation, it was also identified that the official and generated trailers have similar actions. As an example Fig. 10 shows identical actions from three movies in official and generated trailers. Also, it is interesting to note that most of the participants found the style of generated trailers identical to those of certain movie genres.

B. DISCUSSION

This work presents a thumbnail-based approach to facilitate the trailer generation process using client device computational resources. Instead of entire movies, the proposed method analyzes lightweight thumbnail containers to detect personalized action. Our thumbnail-based implementation does not require heavy computational resources. An animated demo is prepared to show all execution steps explained in Section IV `https://youtu.be/sQDrw84_sn8`.

The results show that the official trailers of two listed movies could not be identified and located on YouTube during the search process. Thus, it is assumed that there is no official trailer available for these listed movies online. In this case, the proposed method can be useful in preparing trailers for those movies for which official trailers are unavailable online. In our a movie's dataset, most of the listed movies have only one official trailer. There is a high chance that a single trailer may give negative effects accordingly [9]. The personalized trailer is highly important, in that it will provide users with an essential message, relevant to the movie, in accordance with their interests.

TABLE 9. Selected official trailers information.

No	Title	Total trailers	YouTube ID	Views
1	89 Trailer: Documentary Recounts Arsenal's Dramatic Title win at Anfield	1	UmUiHof01l4	127,335
2	BOBBY – Official Trailer	1	f1dJSOU-CUk	99,500
3	Bodyline Episode 1 (1984)	0	–	–
4	BONE TOMAHAWK Official Trailer (2016) Western	1	QuGmtoQBPEM	498,587
5	Dakota – Trailer	1	CVeqwq-ZvVI	584
6	Django (1966) – Trailer	2	w8Ge2hmSTbo	1,420,015
7	Django Unchained – Official Trailer (HD)	3	eUdM9vrCbow	16,461,257
8	Goal! The Dream Begins 2005 Trailer [HD]	1	67LM5X9-MHA	154,537
9	Iqbal (2005) Trailer	1	5OIco9k7KcE	121,231
10	KENNY	1	6mA6uA2-Rcw	219,456
11	Lagaan: Once Upon a Time in India Official Trailer	1	oSIGQ0YkFxs	363,132
12	Little Big Man (1970) Trailer	1	7K4l5ZZe4-k	31,832
13	M.S.Dhoni – The Untold Story l Official Trailer l Sushant Singh Rajput l Neeraj Pandey	1	6L6XqWoS8tw	34,649,615
14	Oklahoma! – Trailer	1	V6uD9-aLCps	76,569
15	Pelé: Birth of a Legend Official Trailer 1 (2016) - Rodrigo Santoro, Seu Jorge Movie HD	1	XBrfxHOXsDE	3,239,095
16	Playing Away (1987)	0	–	–
17	Shanghai Noon (2000) Trailer	1	FqHg5fc_0_U	37,543
18	Take The Ball, Pass The Ball: Trailer for documentary on Barcelona's Guardiola years	1	VfKls9Eo1ZI	421,157
19	The Game of Their Lives Trailer	1	1H2fRn8PStw	128,803
20	The Indian Fighter (1955) Original Trailer	1	hWP2Un2Dr5I	2,880
21	The Rider l Official Trailer HD (2017)	2	AlrWRttLTkg	848,745
22	The Tracker Trailer	1	2IV3LvS_M6M	92,368
23	The Train Robbers – Original Theatrical Trailer	1	P0rK5Q-TX-k	4,150
24	True Grit' Trailer HD	2	CUiCu-zuAgM	5,977,606
25	Vengeance Valley Original Trailer	1	j2srpV2RN-4	7,126

This study focused on two genres (Western and Sports) in the trailer generation process. However, it could easily be adapted for other genres by utilizing various datasets. In addition, this approach can be extended beyond facilitated personalized trailers towards video summarization and highlights. Along with reducing server-side computational resources in process personalized trailers, the proposed client-side approach can support a privacy-preserving solution [13] using efficient encryption techniques [54], [55], can analyze human activity recognition [56], and can be adapted in three screen TV solutions [57], [58].

VI. CONCLUSION

This paper presented an innovative client-driven approach to facilitate personalized trailer generation process. Instead of analyzing the entire movie, the proposed approach generates personalized trailers by processing lightweight thumbnail containers. The approach does not require heavy processing to detect personalized actions. Experimental results show that it requires less processing time compared to analyzing the entire movie. The HLS server and client are locally configured to validate the proposed method. Instead of open/close connections for every request/response for thumbnail containers and segments, the system uses the HTTP persistent broad- cast connection to preserve bandwidth and computational resources on both sides. To evaluate the effectiveness of the proposed method, 25 broadcast movies were evaluated with the trailer generation process by locating relevant actions in the western and sports genres. A subjective evaluation was conducted to compare the number of relevant actions present in official and generated trailers. The proposed method can support a privacy-preserving solution, where the user can choose to retain all personal data on the client-side. The client-driven approach can be used to create highly responsive and easily scalable systems. Therefore, this system is designed to support a wide range of client hardware with different computational capabilities and has the flexibility to adapt to different network conditions.

REFERENCES

[1] A. He. (2018). *Time Flies: U.S. Adults Now Spend Nearly Half a Day Interacting With Media*. [Online]. Available: https://www.nielsen.com/

[2] M. Mahoney. (2019). *Just the Stats: Why Should You Leverage Video Marketing*. [Online]. Available: https://www.singlegrain.com/

[3] S. Godbole. (2019). *I Believe Netflix Needs Social Features—And Users Agree!* [Online]. Available: https://medium.com/

[4] G. Irie, T. Satou, A. Kojima, T. Yamasaki, and K. Aizawa, "Automatic trailer generation," in *Proc. 18th ACM Int. Conf. Multimedia (MM)*, 2010, pp. 839–842.

[5] S. Asur and B. A. Huberman, "Predicting the future with social media," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, vol. 1, Aug. 2010, pp. 492–499.

[6] K. M. Johnston, "The coolest way to watch movie trailers in the world trailers in the digital age," *Convergence*, vol. 14, no. 2, pp. 145–160, 2008.

[7] C. B. Stapleton and C. E. Hughes, "Mixed reality and experiential movie trailers: Combining emotions and immersion to innovate entertainment marketing," in *Proc. Int. Conf. Hum.-Comput. Interface Adv. Modeling Simulation (SIMCHI)*, 2005, pp. 23–27.

[8] Wikipedia. (2019). *Trailer (Promotion)*. [Online]. Available: https://en.wikipedia.org/

[9] D. Spy. (2019). *Sonic the Hedgehog Movie has Been Delayed Following Trailer Criticism*. [Online]. Available: https://www.digitalspy.com/

[10] S. Min. (2019). *Coming Soon to Netflix: Movie Trailers Crafted by AI*. [Online]. Available: https://www.cbsnews.com/

[11] J. B. A. Chandrashekar, F. Amat, and T. Jebara. (2018). *Artwork Personalization at Netflix*. [Online]. Available: https://medium.com/

[12] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell, ''Explaining the user experience of recommender systems,'' *User Model. User-Adapted Interact.*, vol. 22, nos. 4–5, pp. 441–504, Oct. 2012.

[13] C. Newell and L. Miller, ''Design and evaluation of a client-side recommender system,'' in *Proc. 7th ACM Conf. Recommender Syst. (RecSys)*, 2013, pp. 473–474.

[14] A. F. Smeaton, B. Lehane, N. E. O'Connor, C. Brady, and G. Craig, ''Automatically selecting shots for action movie trailers,'' in *Proc. 8th ACM Int. workshop Multimedia Inf. Retr. (MIR)*, 2006, pp. 231–238.

[15] J. R. Smith, D. Joshi, B. Huet, W. Hsu, and J. Cota, ''Harnessing ai for augmenting creativity: Application to movie trailer creation,'' in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1799–1808.

[16] Y. Kawai, H. Sumiyoshi, and N. Yagi, ''Automated production of TV program trailer using electronic program guide,'' in *Proc. 6th ACM Int. Conf. Image Video Retr. (CIVR)*, 2007, pp. 49–56.

[17] Y. Gong and X. Liu, ''Video summarization using singular value decomposition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2000, pp. 174–180.

[18] Y. Li, T. Zhang, and D. Tretter, ''An overview of video abstraction techniques,'' HP Laboratory, Palo Alto, CA, USA, Tech. Rep. HPL-2001-191, 2001.

[19] R. Kannan, G. Ghinea, and S. Swaminathan, ''What do you wish to see? A summarization system for movies based on user preferences,'' *Inf. Process. Manage.*, vol. 51, no. 3, pp. 286–305, May 2015.

[20] P. Koutras, A. Zlatintsi, E. Iosif, A. Katsamanis, P. Maragos, and A. Potamianos, ''Predicting audio-visual salient events based on visual, audio and text modalities for movie summarization,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 4361–4365.

[21] J. Sang and C. Xu, ''Character-based movie summarization,'' in *Proc. Int. Conf. Multimedia (MM)*, 2010, pp. 855–858.

[22] L. Zhuang, F. Jing, and X.-Y. Zhu, ''Movie review mining and summarization,'' in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage. - CIKM*, 2006, pp. 43–50.

[23] B. Han, J. Hamm, and J. Sim, ''Personalized video summarization with human in the loop,'' in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2011, pp. 51–57.

[24] F. Chen, C. De Vleeschouwer, and A. Cavallaro, ''Resource allocation for personalized video summarization,'' *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 455–469, Feb. 2013.

[25] J. Hannon, K. McCarthy, J. Lynch, and B. Smyth, ''Personalized and automatic social summarization of events in video,'' in *Proc. 15th Int. Conf. Intell. User Interface (IUI)*, 2011, pp. 335–338.

[26] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang, ''A generic framework of user attention model and its application in video summarization,'' *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 907–919, Oct. 2005.

[27] H. Xu, Y. Zhen, and H. Zha, ''Trailer generation via a point process-based visual attractiveness model,'' in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015.

[28] J. You, G. Liu, L. Sun, and H. Li, ''A multiple visual models based perceptive analysis framework for multilevel video summarization,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 3, pp. 273–285, Mar. 2007.

[29] W. Jiang, C. Cotton, and A. C. Loui, ''Automatic consumer video summarization by audio and visual analysis,'' in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2011, pp. 1–6.

[30] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu, ''Video summarization via transferrable structured learning,'' in *Proc. 20th Int. Conf. World Wide Web (WWW)*, 2011, pp. 287–296.

[31] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, and T.-S. Chua, ''Event driven Web video summarization by tag localization and key-shot identification,'' *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 975–985, Aug. 2012.

[32] Z. Wang, M. Kumar, J. Luo, and B. Li, ''Sequence-kernel based sparse representation for amateur video summarization,'' in *Proc. Joint ACM Workshop Modeling Representing Events (J-MRE)*, 2011, pp. 31–36.

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

[34] X. Chen, Y. Zhang, Q. Ai, H. Xu, J. Yan, and Z. Qin, ''Personalized key frame recommendation,'' in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2017, pp. 315–324.

[35] L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, and G. Serra, ''Event detection and recognition for semantic annotation of video,'' *Multimedia Tools Appl.*, vol. 51, no. 1, pp. 279–302, Jan. 2011.

[36] S. Mei, G. Guan, Z. Wang, S. Wan, M. He, and D. Dagan Feng, ''Video summarization via minimum sparse reconstruction,'' *Pattern Recognit.*, vol. 48, no. 2, pp. 522–533, Feb. 2015.

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

[38] T. Yao, T. Mei, and Y. Rui, ''Highlight detection with pairwise deep ranking for first-person video summarization,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 982–990.

[39] N. Babaguchi, Y. Kawai, T. Ogura, and T. Kitahashi, ''Personalized abstraction of broadcasted American football video by highlight selection,'' *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 575–586, Aug. 2004.

[40] P. Dragicevic, G. Ramos, J. Bibliowitcz, D. Nowrouzezahrai, R. Balakrishnan, and K. Singh, ''Video browsing by direct manipulation,'' in *Proc. 26th Annu. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, 2008, pp. 237–246.

[41] W. Hürst, G. Götz, and P. Jarvers, ''Advanced user interfaces for dynamic video browsing,'' in *Proc. 12th Annu. ACM Int. Conf. Multimedia*, 2004, pp. 742–743.

[42] S. Pongnumkul, J. Wang, G. Ramos, and M. Cohen, ''Content-aware dynamic timeline for video browsing,'' in *Proc. 23nd Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, 2010, pp. 139–142.

[43] W. Yang and M. H. Tsai. (2015). *Improving Youtube Video Thumbnails With Deep Neural Nets*. [Online]. Available: https://ai.googleblog.com/

[44] M. O'Leary, ''IIS IIS IIS and ModSecurity,'' in *Cyber Operations*. Berkeley, CA, USA: Springer, 2019, pp. 789–819.

[45] FFmpeg. (2018). *FFmpeg GitHub Page*. [Online]. Available: https://github.com/FFmpeg/FFmpeg

[46] R. Hopkins, ''Digital terrestrial HDTV for North America: The grand alliance HDTV system,'' *IEEE Trans. Consum. Electron.*, vol. 40, no. 3, pp. 185–198, Aug. 1994.

[47] K. J. Ma and R. Bartos, ''HTTP live streaming bandwidth management using intelligent segment selection,'' in *Proc. IEEE Global Telecommun. Conf. - GLOBECOM*, Dec. 2011, pp. 1–5.

[48] M. Thomson. (2013). *Hypertext Transfer Protocol Version 2*. [Online]. Available: https://tools.ietf.org/id/draft-ietf-httpbis-http2-12.html

[49] C. Mueller, S. Lederer, C. Timmerer, and H. Hellwagner, ''Dynamic adaptive streaming over HTTP/2.0,'' in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2013, pp. 1–6

[50] R. Zurawski, ''The hypertext transfer protocol and uniform resource identifier,'' in *The Industrial Information Technology Handbook*. Boca Raton, FL, USA: CRC Press, 2004, pp. 456–478.

[51] K. Soomro, A. Roshan Zamir, and M. Shah, ''UCF101: A dataset of 101 human actions classes from videos in the wild,'' 2012, *arXiv:1212.0402*. [Online]. Available: http://arxiv.org/abs/1212.0402

[52] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ''Rethinking the inception architecture for computer vision,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[53] Video-Dev. (2018). *Hls.JS Github Page*. [Online]. Available: https://github.com/video-dev/hls.js/

[54] G. Mujtaba, M. Tahir, and M. H. Soomro, ''Energy efficient data encryption techniques in smartphones,'' *Wireless Pers. Commun.*, vol. 106, no. 4, pp. 2023–2035, Jun. 2019.

[55] S. Qiu, G. Xu, H. Ahmad, G. Xu, X. Qiu, and H. Xu, ''An improved lightweight two-factor authentication and key agreement protocol with dynamic identity based on elliptic curve cryptography,'' *TIIS*, vol. 13, no. 2, pp. 978–1002, 2019.

[56] S. Arif, J. Wang, Z. Fei, and F. Hussain, ''Video representation via fusion of static and motion features applied to human activity recognition,'' *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 7, 2019.

[57] E.-S. Ryu and N. Jayant, ''Home gateway for three-screen TV using H.264 SVC and raptor FEC,'' *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1652–1660, Nov. 2011.

[58] E.-S. Ryu and C. Yoo, ''Towards building large scale live media streaming framework for a U-city,'' *Multimedia Tools Appl.*, vol. 37, no. 3, pp. 319–338, May 2008.

GHULAM MUJTABA received the B.S. degree in computer science from the COMSATS Institute of Information Technology, Lahore, Pakistan, in 2013, on fully funded ICT R&D scholarship by the Ministry of IT, and the M.S. degree in computer science from Indus University, Karachi, in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Gachon University, Seongnam, South Korea. Prior to joining Gachon University for the Ph.D. degree, in 2018, he has gain vast academic, technical, and professional experience in numerous top-ranking organizations in Pakistan. His research interests are in the areas of personalized media, multimedia communications, and mobile computing.

EUN-SEOK RYU (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Korea University, Seoul, South Korea, in 1999, 2001, and 2008, respectively. In 2008, he was a Research Professor with the Research Institute for Information and Communication Technology, Korea University. From September 2008 to December 2010, he was a Postdoctoral Research Fellow with the Georgia Centers for Advanced Telecommunications Technology (GCATT), School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. From January 2011 to February 2014, he was a Staff Engineer with InterDigital Labs, San Diego, CA, USA, where he researched and contributed to next generation video coding standards, such as HEVC and SHVC. From March 2015 to August 2019, he was an Assistant Professor with the Department of Computer Engineering, Gachon University, Seongnam, South Korea. He was also a Principal Engineer with Samsung Electronics, Suwon, South Korea, where he led a multimedia team. He is currently an Assistant Professor with the Department of Computer Education, Sungkyunkwan University (SKKU), Seoul, South Korea. His research interest is in the area of multimedia communications, including video source coding and wireless mobile systems.