TECHNICAL REPORT | 2023

IM SCIENCE

Hierarchical Time-aware Approach for Video Summarization

(Submitted – conference – BRACIS)

04

TECHNICAL REPORT | 2023

IM SCIENCE

Hierarchical Time-aware Approach for Video Summarization

Citation details:
Leonardo Cardoso Vilela, Gustavo Oliveira Rocha Gomes, Silvio Jamil F. Guimarães, Zenilton Kleber Gonçalves do Patrocínio Jr. (2023). Hierarchical Time-aware Approach for Video Summarization. Technical Report ImScience/PUC Minas #04/2023.

04

Technical report – ImScience 04/2023

Hierarchical Time-aware Approach for Video Summarization

Leonardo Cardoso Vilela · Gustavo Oliveira Rocha Gomes · Silvio Jamil F. Guimarães · Zenilton K. Gonçalves do Patrocínio Jr.

June 8, 2023

Abstract Video summarization consists of generating a concise video representation that captures all its meaningful information. However, conventional summarization techniques often fall short of capturing all the significant events in a video due to their inability to incorporate the hierarchical structure of the video content. This work proposes an unsupervised method, named **Hie**rarchical **T**ime-**a**ware **Summ**arizer– HieTaSumm, that uses a hierarchical approach for that task. In this regard, hierarchical strategies for video summarization have emerged as a promising solution, in which video content is modeled as a graph to identify keyframes that represent the most relevant information. This approach enables the extraction of the frames that convey the central message of the video, resulting in a more effective and precise summary. Experimental results indicate that the proposed approach has great potential. Specifically, it seems to enhance coherence among different video segments, reducing frame redundancy in the generated summaries, and enhancing the diversity of selected keyframes.

Keywords Video summarization · Hierarchical graph-based clustering · Unsupervised learning.

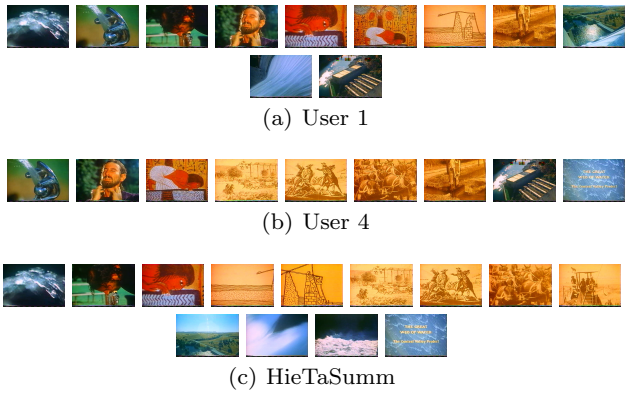
1 Introduction

Video summarization is a challenging task that has gained significant attention in the computer vision and multimedia communities [1,16]. One of the goals of video summarization is to extract essential information from a video and present it in a condensed format [6, 7, 13, 15]. This task is essential for applications such as video captioning, surveillance, synopsis of news videos [10], and video retrieval, among others [1]. The video summarization task involves several sub-tasks, such as keyframe extraction, object tracking, and summarization itself. The keyframe extraction step selects representative frames that capture the essence of the video, while object tracking aims to track important objects across frames [2]. The summarization step involves selecting a subset of keyframes that provide a comprehensive summary of the video while minimizing redundancy. Video summarization techniques can be categorized into unsupervised and supervised approaches, depending on the availability of training data. While unsupervised techniques aim to identify patterns in the video data without any prior knowledge, supervised techniques require labeled data to train the summarization model [1, 16].

Video summarization is particularly useful when dealing with a video collection containing lots of repeated or redundant information spread out over many points in time. In such cases, it becomes a challenge to analyze the entire video and efficiently extract useful information. Video summary techniques can help identify the most important frames in the video that are likely to contain unique and relevant information. By summarizing the video, one can achieve a condensed version that retains the most relevant information while reducing the overall size of the video collection. This allows us to efficiently analyze large video datasets and highlight the most important information, improving the overall effectiveness of video analysis tasks [7, 13].

Figure 1 shows that creating a single summary for a video that accurately reflects every user's perception and preferences can be a challenging task. Since groundtruth data is generated by humans, the interpretation of each user of what is essential and relevant may vary. To generate the groundtruth for the video summarization task, annotators must watch the entire video and identify the most crucial moments. However, what one annotator perceives as essential may differ from another, leading to a subjec-

Silvio Jamil F. Guimarães
ImScience/PUC-Minas – Belo Horizonte 31980-110, Brazil
E-mail: sjamil@pucminas.br



(a) User 1

(b) User 4

(c) HieTaSumm

Fig. 1 Example of the summary generated by the HieTaSumm method compared with the groundtruth. In this case, the HieTaSumm method returns 13 keyframes in contrast to two annotators, the first one (User 1) selects 11 keyframes while the second (User 4) selects 9 keyframes for video $v21$ of the OpenVideo dataset.

tive groundtruth. Hence, the subjectivity of groundtruth generated for each user is a critical aspect to consider in a machine learning method [2, 13].

Regardless of the difficulties related to the subjectivity of groundtruth generated by several users, many unsupervised methods have been proposed over the years. In [8], the authors presented a platform for customizing video summaries. Using clustering techniques, they proposed a method named VISTO, which analyzed low-level features to determine the similarity between frames. Keyframe selection is done by selecting the center of each cluster then a post-processing step is responsible for analyzing and removing possible frame redundancies. In [6], the authors presented a clustering-based strategy to solve the video summarization task named VSUMM. First, a sampling process is made to reduce the number of frames under analysis. Then, frames represented by color histograms were grouped into similar sets by a k-means algorithm. VSUMM results tended to group dispersed frames in time that may have a considerable temporal separation. In [14], the authors presented a graph-based approach for video summarization named HSUMM. The proposed approach was hierarchical and comprised keyframe extraction, scene segmentation, and video summarization stages. During the keyframe extraction stage, their method selected representative frames based on image quality and diversity. In the scene segmentation stage, the video was divided into different scenes based on the visual similarity between frames. Finally, keyframes were combined to generate a video summary. The proposed approach employed a hierarchical graph-based clustering that was capable of generating effective video summaries. In [12], the authors presented an unsupervised approach for summarizing a collection of videos. They developed a diversity-aware optimization method for multi-video summarization by exploring the videos' complementarity.

The video summarization landscape has evolved over the last few years, especially after the introduction of deep learning algorithms. The study in [11] focused on egocentric' video summarization and the challenges of this task. In [3], the authors concentrated on summarization methods that are directly applied to the compressed domain. Finally, the authors in [17] presented the relevant bibliography for dynamic video summarization. According to [1], in deep-learning-based video summarization methods, the video content is represented by deep feature vectors extracted by pre-trained neural networks. The extracted features are then utilized by a deep summarizer network and its output can be either a set of keyframes (i.e., a static summary) or a set of video fragments (that form a dynamic summary).

This work proposes an unsupervised method for video summarization that considers changes in video content over time, named **Hie**rarchical **T**ime-**a**ware **Summ**arizer– HieTaSumm. Similarly to recent deep-learning-based approaches, the proposed method uses pre-trained neural networks to generate video frame descriptions. However, it does not adopt a deep summarizer network to avoid the challenges related to its training. Instead, a hierarchical graph-based clustering strategy is adopted. It is worth mentioning that the proposed method assesses frame importance over time for selecting keyframes that comprise the video summary, which is different from other hierarchical approaches. The major contributions of this work are two-fold: (i) a strategy for video summarization that incorporates frame importance over time for selecting keyframes; and (ii) the identification of keyframes through a hierarchical graph-based clustering using deep-learning-based descriptors and a dynamic strategy to define summary sizes.

This work is organized as follows. Section 2 defines many concepts used in this work. Section 3 presents the proposed method, followed by the experimental results in Section 4. Finally, Section 5 draws some conclusions and future work proposals.

2 Fundamental Concepts

Let $\mathbb{A} \subset \mathbb{N}^2$, $\mathbb{A} = \{0, \ldots, \mathrm{H}-1\} \times \{0, \ldots, \mathrm{W}-1\}$, where H and W are the width and height of each frame, respectively, and, $\mathbb{T} \subset \mathbb{N}$, $\mathbb{T} = \{0, \ldots, N-1\}$, in which N is the number of frames of a video. A frame f is a function from $\mathbb{A}$ to $\mathbb{R}^3$, where for each spatial position (x, y) in $\mathbb{A}$, $f(x, y)$ represents the color value at pixel location (x, y). A video V_N, in domain $\mathbb{A} \times \mathbb{T}$, can be seen as a sequence of frames f. It can be described by $\mathrm{V}_N = (f)_{t \in \mathbb{T}}$, where N is the number of frames contained in the video.

A frame f is usually described in terms of a global descriptor $d(f)$. Let f_{t_1} and f_{t_2} be two video frames at locations t_1 and t_2, respectively. The (dis)similarity between f_{t_1} and

f_{t_2} can be evaluated by a distance measure $\mathcal{D}(d(f_{t_1}), d(f_{t_2}))$ between their descriptors. There are several choices for $\mathcal{D}(d(f_{t_1}), d(f_{t_2}))$, i.e., the distance measure between two frames depending on the global descriptor, e.g. histogram/frame difference, histogram intersection, difference of histograms means, and even the L_2 norm.

A time-aware frame similarity graph $G_\delta = (V, E_\delta)$ is a weighted undirected graph. Each node $v_t \in V$ represents a frame $f_t \in \mathrm{V}_N$. There is an edge $e \in E_\delta$ with a weight $w(e) = \mathcal{D}(d(f_{t_1}), d(f_{t_2}))$ between two nodes v_{t_1} and v_{t_2} if the difference between their time indexes falls below a specified threshold δ, i.e.,

$$E_\delta = \{\ (v_{t_1}, v_{t_2}, \mathcal{D}(d(f_{t_1}), d(f_{t_2}))) \mid v_{t_1}, v_{t_2} \in V, v_{t_1} \neq v_{t_2}, |t_2 - t_1| \leq \delta\}. \quad (1)$$

This constraint over the frames' time indexes limits the connections between distant video frames, effectively allowing the proposed method to consider two frames as similar only if they are not very far in time. This is a noteworthy distinction from many other approaches in the literature, which may consider two frames as similar independently from their time occurrence. Doing that permits the proposed method to assess frame importance over time for selecting it as a keyframe to form the video summary even when it seems to reoccur throughout the video. Figure 2(a) illustrates a time-aware frame similarity graph with $\delta = 4$.

Similar to [14], this work also constructs a hierarchy based on a minimum spanning tree (MST) of the original graph. So, we define an edge-weighted tree of frames $T_{G_\delta} = (V, E^*_\delta)$ is a connected acyclic subgraph of G_δ, i.e., $E^*_\delta \subseteq E_\delta$. The weight of T_{G_δ} is equal to the sum of weights of all edges belonging to E^*_δ, i.e., $w(T_{G_\delta}) = \sum_{e \in E^*_\delta} w(e)$. The minimum spanning tree of frames $T^*_{G_\delta}$ is a tree of frames whose weight is minimal.

Given a finite set V, a *partition* of V is a set $\mathbf{P}$ of nonempty disjoint subsets of V whose union is V. Any element of $\mathbf{P}$, denoted by $\mathbf{R}$, is called a *region* of $\mathbf{P}$. Given two partitions $\mathbf{P}$ and $\mathbf{P'}$ of V, $\mathbf{P'}$ is said to be a (total) refinement of $\mathbf{P}$, denoted by $\mathbf{P'} \preceq \mathbf{P}$, if any region of $\mathbf{P'}$ is included in a region of $\mathbf{P}$. Let $\mathcal{H} = (\mathbf{P}_1, \ldots, \mathbf{P}_\ell)$ be a set of ℓ partitions on V. $\mathcal{H}$ is a hierarchy if $\mathbf{P}_{i-1} \preceq \mathbf{P}_i$, for any $i \in \{2, \ldots, \ell\}$. According to [5], an MST can be utilized to represent a hierarchy, and a weighted MST of a graph can address any connected hierarchy for that graph. Additionally, the work in [9] demonstrated that creating a hierarchical graph segmentation involves reweighting an MST using a dissimilarity measure between regions. Thus, the proposed method utilizes an MST of frame similarity graph $T^*_{G_\delta}$ to obtain a hierarchy $\mathcal{H}$ which is then used to obtain frame clusters.

Finally, a hierarchical segmentation of G_δ into k components is equivalent to the partition of a hierarchy $\mathcal{H}$ into k regions (containing more similar elements) and can be done by removing $k-1$ edges that present higher weights (representing greater dissimilar) from the $T^*_{G_\delta}$ (since it represents $\mathcal{H}$). This strategy incorporates a similarity measure between clusters while partitioning the graph, providing a more comprehensive approach than traditional methods that only consider the similarity between isolated frames.

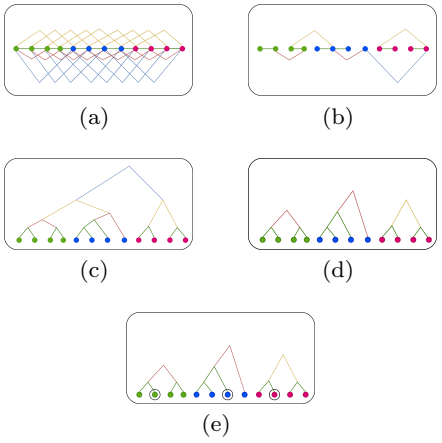


Fig. 2 Illustration of the proposed method steps: (a) generation of a time-aware frame similarity graph G_δ for a video; (b) computation of its minimum spanning tree $T^*_{G_\delta}$; (c) creation of a hierarchy $\mathcal{H}$ based on $T^*_{G_\delta}$; (d) generation of subsets of frames through hierarchy cuts (edge removals); and (e) selection of keyframes to represent each subset. These keyframes are the result of the summarization process.

3 Hierarchical Time-Aware Video Summarization

Figure 2 illustrates the proposed method steps. The main steps of HieTaSumm method are the following: (i) generation of a time-aware frame similarity graph G_δ to represent a video; (b) computation of a minimum spanning tree $T^*_{G_\delta}$ for that graph; (c) creation of a hierarchy $\mathcal{H}$ based on the $T^*_{G_\delta}$; (d) generation of subsets of frames through cuts on the hierarchy; and (e) selection of keyframes to represent each subset.

The HieTaSumm method (see Algorithm 1) created and uses a frame similarity graph G_δ. Each vertex represents a distinct video frame and there is an edge between two vertices if the difference between their time indexes falls below a specified threshold δ. Equation 2 represents this constraint and is implemented at line 7 of Algorithm 1.

$$|t_2 - t_1| < \delta_t \quad (2)$$

in which t_f and $t_{f'}$ represent the time indexes of frames f and f', respectively. Additionally, the edge weight represents the (dis)similarity between frames.

The proposed method employs the Kruskal algorithm to obtain the MST $T^*_{G_\delta}$ from G_δ, while the *watershed by*

Algorithm 1 Hierarchical time-aware video summarization

Input: A video V_N, threshold value δ
Output: A list of keyframes $\mathcal{K}$

Create a graph G_δ with a vertice set $V = \emptyset$ and an edge set $E_\delta = \emptyset$
for all $f_t \in V_N$ **do**
  $V := V \cup \{f\}$ // Insert f in V if f does not belong to it
  $d(f_t) := GenerateDescriptor(f_t)$ // Obtain a descriptor for frame f_t
end for
for all $f_{t_1} \in V_N$ **do**
  for all $f_{t_2} \in V_N$ such that $f_{t_2} \neq f_{t_1}$ and $|t_2 - t_1| < \delta$ **do**
    $w = \mathcal{D}(d(f_{t_1}), d(f_{t_2}))$
    $G.AddEdge(f_{t_1}, f_{t_2}, w)$ // Insert edge (f_{t_1}, f_{t_2}) with (dis)similarity as weight
  end for
end for
$T^*_{G_\delta} := G_\delta.Obtain\_MST\_from\_Graph()$
$\mathcal{H} := T^*_{G_\delta}.Generate\_Hierarchy\_from\_MST()$
$\mathcal{K} := \mathcal{H}.Dynamic\_Selection\_of\_Keyframes()$ // Remove edges from $\mathcal{H}$ to obtain // frame sets and select the central // vertice of each set as keyframe
Return $\mathcal{K}$;

area [4] is used to generate a hierarchy $\mathcal{H}$ from $T^*_{G_\delta}$. Once a hierarchy $\mathcal{H}$ is constructed, a hierarchical segmentation of G_δ generates a video summary of size k. For that, The proposed method needs only to remove the $k-1$ edges with higher weights from $\mathcal{H}$. Instead of generating a fixed-size video summary, we adopt a strategy for identifying the moment when stability is reached during the edge removal process that is similar (but distinct) to the one used in [14]. Let e' be the edge with the highest weight in the hierarchy $\mathcal{H}$. Thus, the edge e' is removed only when its weight $w(e')$ is greater than or equal to an equilibrium measure function $F(e')$, i.e., $w(e') \geq \mathbf{F}(e)$. In this work, the equilibrium measure function is given by Equation 3.

$$\mathbf{F}(e) = \gamma \sigma_w(e) \tag{3}$$

in which $\sigma_w(e)$ represents the standard deviation of all edge weights of the connected component that contains edge e, and γ is a parameter related to the allowed variability. During tests, we have set γ empirically.

Finally, after dividing the hierarchy into several connected components, central frames (concerning chronological order) are selected as keyframes for the video summary.

This dynamic choice of the number of components and, consequently, the size of the video summary becomes essential when it comes to videos that contain numerous very similar scenes. In such cases, employing a static number of frames for all videos can result in redundant and repetitive content in the summary. By adopting a dynamic approach, the method can infer an adequate summary size based on the specific video content and characteristics.

4 Experimental Results

This section provides a comprehensive analysis of results obtained by the proposed approach to video summarization with a dynamic selection of video summaries.

We compared HieTaSumm with other unsupervised video summarization methods, namely HSUMM [14], VSUMM1 [6], VSUMM2 [6], VISTO [8] and Open Video summaries (referred to like OVSummary). These comparative assessments allow for a comprehensive review of the performance and effectiveness of HieTaSumm against these established approaches.

4.1 Implementation Details and Dataset

Similar to [8, 14], we applied the proposed method to the same collections of videos from the OpenVideo dataset (referred to as the VSUMM dataset in [16]). This dataset contains 50 videos of different genres. All videos are in MPEG-1 format (30 fps, 352×240 pixels). The genres are distributed into documentary, educational, ephemeral, historical, and lecture. The time duration of each video varies from 01 to 04 minutes. The process of creating of user summary consists of the collaboration of 50 different persons. Each user is dealing with the task of choosing the keyframes for 5 videos. Thus, 250 were created for the dataset each video has 05 different user summaries generated manually. And, as a way to pre-process the video dataset we extracted 04 fps from all videos.

For the creation of the frame similarity graph, we use ResNet50 and VGG16 (both pre-trained on ImageNet) to extract frame descriptors. The cosine similarity was used to assess the similarity between two frame descriptors. And, we also set $\delta = 32$ (i.e., 08 seconds with 04 fps) and $\gamma = 0.05$, during the experiments.

4.2 Evaluation Metrics

Assessing frame quality in the context of video summarization poses a distinct challenge because of the many ways in which frames can be constructed while conveying similar meanings. These variations can arise from using different analyzes of resources from different informational aspects. Although humans have an intuitive understanding of this process, abstract evaluation remains an open question without a specific framework. As a result, the conventional practice involves adapting similar metrics that have been stretched to accommodate the specific requirements of the video summary task. By re-purposing and customizing these metrics, researchers, and practitioners can assess the

effectiveness and fidelity of summaries generated in video summarization, despite the inherent complexities and subjectivity involved in sentence evaluation [6,14].

In order to compute the improvement of the frame selection, we will evaluate the obtained results following the same approach used by the authors of [6,14]. They reported their results using metrics widely disseminated in the literature such as CUSa, CUSe [6,14], and COV [14], defined by the Equations 4–6, respectively, with the objective of evaluating the similarity between the frames generated by their summarization method and the GT results.

$$\text{CUSa} = \frac{m_A}{n_U} \tag{4}$$

$$\text{CUSe} = \frac{\overline{m}_A}{n_U} \tag{5}$$

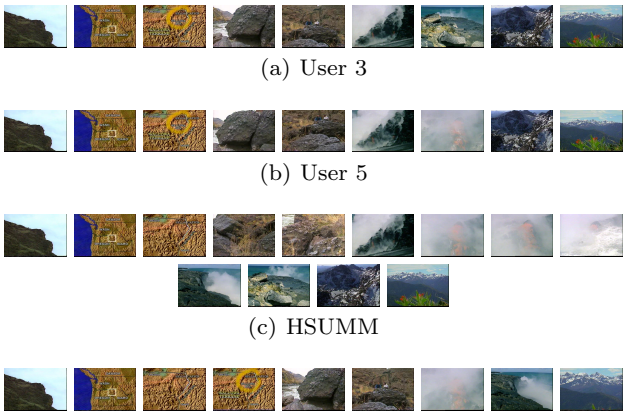
in which m_A denotes the number of matching keyframes generated from the Automatic Summary (AS), $\overline{m}_A$ represent non-matching keyframes from AS, and n_U are the number of keyframes selected for the user to represent the user summary (U) to each video.

$$\text{COV} = \frac{\sum_{U \in US} |M(AS, U)|}{\sum_{U \in US} |U|} \tag{6}$$

in which $M(X, Y)$ and $|.|$ are the maximum matching between two sets of different elements X and Y, and the cardinality of a set, respectively.

While those two first metrics provide valuable insights, they often fail to measure the diversity displayed in user summaries as COV does. Furthermore, the calculation of averages for each user's measurements can introduce distortions and inaccuracies. Specifically, the CUSa, which is commonly employed to assess user opinions, fails to effectively capture the diversity of these opinions. To illustrate, consider two users, A and B, providing summaries for the same video. Let the summary of user A be $U_A = \{X, Y\}$ while the summary of user B is $U_B = \{M, N, O, P, Q, R, S, T, U, V\}$, in which each character denotes a single frame of video. Now suppose that three distinct methods generate summaries: $AS_1 = \{X, Y\}$, $AS_2 = \{M, N, O, P, Q, R, S, T, U, V\}$, and $AS_3 = \{X, M, N, O, P, Q\}$. Despite these summaries being completely different, they provide the same accuracy rate (i.e., CUSa = 0.5). This highlights the limitations of CUSa in accurately assessing divergence of opinion and the need for more comprehensive assessment metrics [6,14].

Unlike CUSa, COV assesses the extent to which an automatic summary covers all user-generated summaries. This measure takes into account both the diversity of opinions expressed by users and the degree of agreement among them. Specifically, the CUSa measure calculates the average ratio between each user's summary and an automatic summary, thus capturing the level of agreement between the two. In contrast, COV assesses the proportion of an automatic summary that aligns with all user summaries, providing a measure of overall covering. We use COV as the first metric to compute the effectiveness of the HieTaSumm. The reader should refer to [6,14] for more information about those metrics.



(a) User 3

(b) User 5

(c) HSUMM

(d) HieTaSumm

Fig. 3 Comparative example of HieTaSumm results compared with HSUMM results and with the frames selected by the User 3 and User 5 (both selected 9 frames). The video summary generated by HieTaSumm contains 9 frames.

4.3 Quantitative Analysis

Table 1 presents the HieTaSumm results. We used ResNet50 and VGG16 to extract frame descriptors for the construction of the frame similarity graph. During the evaluation of the results, we also used ResNet50 and VGG16 to extract frame descriptors but the cosine similarity was used to verify the agreement between the groundtruth and automatic summaries. We have also used color histograms (CH) during the assessment of the results.

Table 1 presents the average values of all metrics for the 50 videos belonging to the dataset. The results are presented for different levels of precision (between groundtruth and automatic summaries). It is possible to notice that the use of ResNet50 presents a slight improvement compared to the results with VGG16 (under a greater precision in evaluation), and the VGG16 presented better results (under a lower preciseness in evaluation). Moreover, it is also possible to observe the high values of COV and CUSa achieved by HieTaSumm method, and even under a higher precision in evaluation, the proposed method still presents competitive results.

4.4 Qualitative Analysis

To provide a better understanding of the results obtained and their improvements, Figures 3 and 4 present samples of summaries generated by various approaches in the literature, including HSUMM [14], VSUMM1 [6], VSUMM2 [6], VISTO [8] and Open Video summaries (referred to like OVSummary), and the groundtruth (GT) results, alongside

Table 1 Performance of HieTaSumm method for different levels of precision in evaluation of video summaries. CUSa, CUSe, and COV values were multiplied by 10^2 to improve readability.

Metrics	Precision of Matches (%)												
($\times 10^2$)	100	99	98	95	90	85	80	75	70	65	60	55	50
ResNet50 + CH													
COV	23.09	35.48	41.81	56.55	68.87	77.87	84.53	87.73	89.55	89.90	90.16	90.33	90.42
CUSa	23.27	35.91	42.42	57.17	69.64	78.90	85.75	88.89	90.68	91.01	91.26	91.46	91.54
CUSe	76.73	64.10	57.58	42.83	30.36	21.11	14.25	11.11	09.32	08.99	08.74	8.54	8.46
VGG16 + CH													
COV	22.85	35.38	42.20	56.74	67.86	77.58	85.65	88.35	89.21	90.08	90.27	90.33	90.42
CUSa	23.01	35.59	42.64	56.34	68.54	78.52	86.85	89.48	90.34	91.22	91.40	91.46	91.54
CUSe	76.99	64.41	57.36	43.66	31.46	21.47	13.15	10.52	09.66	08.78	08.60	08.54	08.46
ResNet50 + ResNet50													
COV	08.29	15.66	20.32	30.67	41.93	49.28	53.66	57.64	60.39	63.16	65.24	67.94	70.60
CUSa	08.45	15.90	20.58	31.12	42.20	49.68	54.17	58.22	60.91	63.70	65.82	68.50	71.30
CUSe	91.55	84.10	79.42	68.88	57.80	50.32	45.83	41.77	39.09	36.30	34.18	31.50	28.70
VGG16 + VGG16													
COV	01.43	12.66	20.90	35.21	49.35	57.92	63.57	69.63	74.92	76.96	78.78	81.24	82.95
CUSa	01.60	12.66	20.93	35.43	50.00	58.32	64.02	70.16	75.38	77.54	79.36	81.89	83.55
CUSe	98.40	87.34	79.07	64.57	50.00	41.68	35.98	29.84	24.62	22.46	20.64	18.11	16.45

those generated by the HieTaSumm method. This comparison enables the evaluation of time awareness, similarity with the GT results, and the rate of the frames selected by the HieTaSumm method and others.

Figure 3 shows the results generated by the HieTaSumm method along with HSUMM [14] results, and the summaries generated by two users. Each frame list created for each user encapsulates a distinct selection of frames, reflecting individual preferences and perspectives. Employing cosine similarity, we can quantify the degree of similarity between the GT of the users and that generated for HieTaSumm method. However, it is essential to recognize that similarity is subjective and may vary among observers. Factors such as the weighting of different frames, the level of granularity in frame selection, and the specific context of the video all influence perceived similarity. Therefore, when evaluating the cosine similarity between two lists of frames, it is crucial to consider the subjective nature of the perception and the different perspectives that individuals bring to the comparison. The result obtained for the HSUMM has a much higher number of frames than the others and, due to this, they present a large number of frames with high similarity. Furthermore, HSUMM results may not preserve chronological order.

On the other hand, HieTaSumm method presents a fluid and coherent result. Furthermore, the select keyframes are very similar to those frames in GT. For all keyframes selected by HieTaSumm method, only one frame does not have another directly correlated with those selected by the two users. But, in all cases, even with different keyframes selected by the users, the automatic summary generated by HieTaSumm method is very close to theirs (especially for Users 3 and 5 shown in Figure 3). In addition, the unrelated keyframe preserves temporal order and, when we look at the three keyframes in which the map is present, it is possible to observe that a refinement process takes place to identify the correct highlighted region, starting from a global visualization to an analysis local that identifies the region in focus as the most important point of location on the map of the region presented in the video.

Figure 4 also presents some subjective characteristics for the keyframes selected by users 2 and 3. Considering the number of frames selected, 15 and 17 respectively, it tends to suggest the existence of a larger number of scene modifications. This variation can cause the selection of a greater number of frames returned by automatic methods, but the increase in the number of scenes can cause frames to be repeated by automatic methods. In this way, the returned summaries have a great challenge of maintaining temporal coherence, but without two highly similar frames being selected without the presence of other events. With this difficulty in mind, OVSummary presents a series of repeated frames side by side. Seen displays some repeated frames, but a reduced number of frames with more similar information. VSUMM1 observes more scene modification and has some information that tends to be more similar. VSUMM2 tends to keep the results without redundancy but without the presence of some scenes more relevant to the user. Finally, the hierarchical approach used by HieTaSumm tends to reduce the redundancy of information with a lot of similarity. HieTaSumm results has a smaller number of keyframes, but these keyframes are more related to user summaries. Moreover, keyframes selected by HieTaSumm method keep the temporal ordering and shows that the dynamic selection of summary size helps to better capture the changing scenes more smoothly.

(a) User 2

(b) User 3

(c) OVSummary

(d) Visto

(e) VSUMM1

(f) VSUMM2

(g) HieTaSumm

Fig. 4 Comparative example of HieTaSumm results compared with the results of VSUMM1 [6], VSUMM2 [6], VISTO [8], OVSummary and with the frames selected by the User 2 and User 3.

5 Conclusion

This work proposes an unsupervised method for video summarization that considers changes in video content over time, named **Hie**rarchical **T**ime-**a**ware **Summ**arizer– HieTaSumm. It uses pre-trained neural networks to generate video frame descriptions with a hierarchical graph-based clustering strategy. The proposed method explores a time-aware frame similarity graph to represent video content considering changes over time. Moreover, a dynamic strategy for defining summary size is adopted. Experimental results indicate that the proposed approach has great potential. Specifically, it seems to enhance coherence among different video segments, reducing frame redundancy in the generated summaries, and enhancing the diversity of selected keyframes.

Future works may explore other strategies for selecting keyframes and different hierarchies. It might also be interesting to investigate the impact of different datasets with little scene modifications. Following these future research directions, we can advance the video summary field and further refine the dynamic frame selection approach to provide more accurate, informative, and user-centric video summaries.

References

1. Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V., Patras, I.: Video summarization using deep neural networks: A survey. Proceedings of the IEEE **109**(11), 1838–1863 (2021)
2. Asha Paul, M.K., Kavitha, J., Jansi Rani, P.A.: Key-frame extraction techniques: A review. Recent Patents on Computer Science **11**(1), 3–16 (2018)
3. Basavarajaiah, M., Sharma, P.: Survey of compressed domain video summarization techniques. ACM Comput. Surv. **52**(6) (2019)
4. Cousty, J., Najman, L.: Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. In: Processing: 10th International Symposium Mathematical Morphology, ISMM 2011, pp. 272–283 (2011)
5. Cousty, J., Najman, L., Kenmochi, Y., Guimarães, S.: Hierarchical segmentations with graphs: Quasi-flat zones, minimum spanning trees, and saliency maps. J. Math. Imaging Vis. **60**(4), 479–502 (2018)
6. De Avila, S.E.F., Lopes, A.P.B., da Luz Jr, A., de Albuquerque Araújo, A.: Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recognition Letters **32**(1), 56–68 (2011)
7. Ejaz, N., Tariq, T.B., Baik, S.W.: Adaptive key frame extraction for video summarization using an aggregation mechanism. Journal of Visual Communication and Image Representation **23**(7), 1031–1040 (2012)
8. Furini, M., Geraci, F., Montangero, M., Pellegrini, M.: Visto: visual storyboard for web video browsing. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, pp. 635–642 (2007)
9. Guimarães, S., Kenmochi, Y., Cousty, J., Patrocinio, Z., Najman, L.: Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity: the case of the felzenszwalb-huttenlocher method. Mathematical Morphology-Theory and Applications **2**(1), 55–75 (2017)
10. Lu, G., Zhou, Y., Li, X., Yan, P.: Unsupervised, efficient and scalable key-frame selection for automatic summarization of surveillance videos. Multimedia Tools and Applications **76**, 6309–6331 (2017)
11. del Molino, A.G., Tan, C., Lim, J.H., Tan, A.H.: Summarization of egocentric videos: A comprehensive survey. IEEE Transactions on Human-Machine Systems **47**(1), 65–76 (2017)
12. Panda, R., Mithun, N.C., Roy-Chowdhury, A.K.: Diversity-aware multi-video summarization. IEEE Transactions on Image Processing **26**(10), 4712–4724 (2017)
13. Pandey, S., Dwivedy, P., Meena, S., Potnis, A.: A survey on key frame extraction methods of a mpeg video. In: 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 1192–1196 (2017)
14. dos Santos Belo, L., Caetano Jr, C.A., do Patrocínio Jr, Z.K.G., Guimarães, S.J.F.: Summarizing video sequence using a graph-based hierarchical approach. Neurocomputing **173**, 1001–1016 (2016)

15. Song, Y., Vallmitjana, J., Stent, A., Jaimes, A.: Tvsum: Summarizing web videos using titles. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), pp. 5179–5187 (2015)
16. Tiwari, V., Bhatnagar, C.: A survey of recent work on video summarization: approaches and techniques. Multimedia Tools and Applications **80**(18), 27,187–27,221 (2021)
17. Vivekraj, V., Debashis, S., Balasubramanian, R.: Video skimming: Taxonomy and comprehensive survey. ACM Comput. Surv. **52**(5) (2019)